Chapter 5
Configure Physical Interface Properties

The software driver for each network media type sets reasonable default values for general interface properties, such as the interface's MTU size, receive and transmit leaky bucket properties, link operational mode, and clock source. To modify any of the default general interface properties, include one or more statements in the [edit interfaces *interface-name*] hierarchy:

```
interfaces {
    interface-name {
        accounting-profile name;
        aggregated-ether-options {
            aggregated-ether-interface-options;
        }
        aggregated-sonet-options {
            aggregated-sonet-interface-options;
        }
        atm-options {
            atm-interface-options;
        }
        clocking clock-source;
        dce;
        disable;
        description text;
        e1-options {
            e1-interface-options;
        }
        e3-options {
            e3-interface-options;
        }
        encapsulation type;
        fastether-options {
            fastether-interface-options;
        }
        gigether-options {
            gigether-interface-options;
            }
        hold-time up milliseconds down milliseconds;
        keepalives <down-count number> <interval seconds> <up-count number>;
        link-mode mode;
```

```
            lmi {
                lmi-type (ansi | itu);
                n391dte number;
                n392dce number;
                n392dte number;
                n393dce number;
                n393dte number;
                t391dte seconds;
                t392dce seconds;
            }
            mac mac-address;
            mtu bytes;
            multiservice-options {
                boot-command filename
                (core-dump | no-core-dump);
                (syslog | no-syslog);
            }
            no-keepalives;
            no-traps;
            ppp-options {
                chap {
                    access-profile name;
                    local-name name;
                    passive;
                }
            }
            receive-bucket {
                overflow (tag | discard);
                rate percentage;
                threshold number;
            }
            sonet-options {
                sonet-interface-options;
            }
            speed (10m | 100m);
            t1-options {
                t1-interface-options;
            }
            t3-options {
                t3-interface-options;
            }
            traceoptions {
                flag flag <flag-modifier> <disable>;
            }
            transmit-bucket {
                overflow (discard);
                rate percentage;
                threshold number;
            }
            unit {
                logical-interface-statements;
            }
            vlan-tagging;
        }
    }
```

This chapter discusses the following physical interface properties that you can configure:

Table 2 lists statements that you can use to configure physical interfaces.

Table 2: Statements for Physical Interface Properties

Statement	Interface Types	Usage Guidelines
access-profile *name*	Interfaces with PPP encapsulation.	"Configure PPP Challenge Handshake Authentication Protocol" on page 47.
accounting-profile *name*	All.	"Configure Accounting" on page 54.
aggregated-ether-options	Aggregated Ethernet interfaces.	"Configure Aggregated Ethernet Interfaces" on page 59.
aggregated-sonet-options	Aggregated SONET/SDH interfaces.	"Configure Aggregated SONET/SDH Interfaces" on page 94.
atm-options	ATM interfaces.	"Configure ATM Physical Interface Properties" on page 60.
boot-command *filename*	Passive monitoring interfaces.	"Configure Multiservice Physical Interface Properties" on page 81.
chap	Interfaces with PPP encapsulation.	"Configure PPP Challenge Handshake Authentication Protocol" on page 47.
clocking *clock-source*	ATM, DS-0, E1, E3, SONET/SDH, T1, and T3 interfaces.	"Configure the Clock Source" on page 52.
(core-dump | no-core-dump)	Passive monitoring interfaces.	"Configure Multiservice Physical Interface Properties" on page 81.
dce	Interfaces with Frame Relay encapsulation.	"Configure the Router as a DCE" on page 53.
description *text*	All.	"Add an Interface Description to the Configuration" on page 40.
disable	All.	"Disable a Physical Interface" on page 58.
e1-options	E1 interfaces.	"Configure E1 and T1 Physical Interface Properties" on page 64.
e3-options	E3 interfaces.	"Configure E3 and T3 Physical Interface Properties" on page 70.
encapsulation *type*	All interface types except aggregated Ethernet, loopback, and multicast tunnel.	"Configure Interface Encapsulation" on page 44.
fastether-options	Fast Ethernet interfaces.	"Configure Ethernet Physical Interface Properties" on page 77.
gigether-options	Gigabit Ethernet interfaces.	"Configure Ethernet Physical Interface Properties" on page 77.
hold-time up *milliseconds* down *milliseconds*	All interface types except aggregated SONET/SDH, GRE tunnel, and IP tunnel.	"Damp Interface Transitions" on page 57.
keepalives <down-count *number*> <interval *seconds*> <up-count *number*>	Aggregated SONET/SDH, DS-0, E1, E3, SONET, T1, and T3 interfaces.	"Configure Keepalives" on page 50.
link-mode *mode*	Management Ethernet (fxp0) and Fast Ethernet interfaces.	"Configure the Link Characteristics" on page 40.
lmi	Interfaces with Frame Relay encapsulation.	"Configure Keepalive Settings on Frame Relay LMI" on page 51.
lmi-type (ansi | itu)	Interfaces with Frame Relay encapsulation.	"Configure Keepalive Settings on Frame Relay LMI" on page 51.

Statement	Interface Types	Usage Guidelines
local-name *name*	Interfaces with PPP encapsulation.	"Configure PPP Challenge Handshake Authentication Protocol" on page 47.
mac *mac-address*	Management Ethernet interface (fxp0).	"Configure the MAC Address on the Management Ethernet Interface" on page 50.
mtu *bytes*	All interface types except Management Ethernet (fxp0), loopback, multilink, and multicast tunnel.	"Configure the Media MTU" on page 41.
multiservice-options	Passive monitoring interfaces.	"Configure Multiservice Physical Interface Properties" on page 81.
no-keepalives	Interfaces with PPP, Frame Relay, or Cisco HDLC encapsulation.	"Configure Keepalives" on page 50.
no-traps	All.	"Disable SNMP Notifications on Physical Devices" on page 58.
passive	Interfaces with PPP encapsulation.	"Configure PPP Challenge Handshake Authentication Protocol" on page 47.
ppp-options	Interfaces with PPP encapsulation.	"Configure PPP Challenge Handshake Authentication Protocol" on page 47.
receive-bucket	All interface types except ATM, Fast Ethernet, and Gigabit Ethernet.	"Configure Receive and Transmit Leaky Bucket Properties" on page 53.
sonet-options	SONET interfaces.	"Configure SONET/SDH Physical Interface Properties" on page 82.
speed (10m | 100m)	Management Ethernet interface (fxp0) and the Fast Ethernet 12-port and 48-port PIC.	"Configure the Interface Speed" on page 49.
(syslog | no-syslog)	Passive monitoring interfaces.	"Configure Multiservice Physical Interface Properties" on page 81.
t1-options	T1 interfaces.	"Configure E1 and T1 Physical Interface Properties" on page 64.
t3-options	T3 interfaces.	"Configure E3 and T3 Physical Interface Properties" on page 70.
traceoptions	All.	"Trace Operations of an Individual Router Interface" on page 57.
transmit-bucket	All interface types except ATM, Fast Ethernet, and Gigabit Ethernet.	"Configure Receive and Transmit Leaky Bucket Properties" on page 53.
vlan-tagging	Fast Ethernet and Gigabit Ethernet interfaces.	"Configure 802.1Q VLAN Tagging" on page 98.

Configure the Interface Name

Each interface has a name that identifies the physical interface type and the location of the interface card in the chassis. To configure the interface name, specify it at the [edit interfaces] hierarchy level:

```
[edit]
interfaces {
    interface-name {
        ...
```

}

Specify the interface name in the following format:

physical<:*channel*>.*logical*

Channelized interfaces have some special requirements; For more information, see “Configure Channelized Interfaces” on page 191.

The physical part of an interface name has the following format:

type-fpc/*pic*/*port*

type is the media type and can be one of the following:

ae—Aggregated Ethernet interface. This is actually a virtual aggregated link and has a different naming format; for more information, see “Configure Aggregated Interfaces” on page 39.

as—Aggregated SONET/SDH interface. This is actually a virtual aggregated link and has a different naming format; for more information, see “Configure Aggregated Interfaces” on page 39.

at—ATM interface.

ds—DS-0 interface. You can configure a Channelized DS-3 to DS-0 PIC or a Channelized E1 PIC with this interface type.

e1—E1 interface. If the name does not include a channel identifier, it is assumed to be a copper-cable-based E1 interface. If the name includes a channel identifier, it is assumed to be an STM-1 channel on a Channelized STM-1 to E1 interface.

e3—E3 interface.

es—Encryption interface.

fe—Fast Ethernet interface.

fxp—Management and internal Ethernet interfaces.

ge—Gigabit Ethernet interface.

gr—Generic Route Encapsulation tunnel interface.

ip—IP-over-IP encapsulation tunnel interface.

lo—Loopback interface.

ml—Multilink interface.

mo—Passive monitoring interface.

mt—Multicast tunnel interface.

so—SONET interface.

t1—T1 interface. If the name does not include a channel identifier, it is assumed to be a copper-cable-based T1 interface. If the name includes a channel identifier, it is assumed to be a DS-1 channel on a Channelized DS-3 interface or a Channelized OC-3 to T1 interface.

t3—T3 interface. If the name does not include a channel identifier, it is assumed to be a copper-cable-based T3 interface. If the name includes a channel identifier, it is assumed to be a DS-3 channel on a Channelized OC-12 interface.

vt—VPN loopback tunnel interface.

fpc is the slot in which the FPC card is installed.

pic is the number of the PIC location in which the interface card is installed on the FPC.

port is the specific port on a PIC.

The logical unit part of the interface name corresponds to the logical unit number, which can be a number in the range 0 through 16384.

Configure Aggregated Interfaces

You specify aggregated interfaces by assigning a number for the aggregated interface. For aggregated Ethernet interfaces, configure ae*x* as in the following example:

```
[edit interfaces]
ae0 {
…
}
```

For aggregated SONET/SDH interfaces, configure as*x* as in the following example:

```
[edit interfaces]
as0 {
…
}
```

The maximum number of aggregated interfaces is 16, and the assigned number can range from 0 through 15. You should not mix SONET and SDH mode on the same aggregated interface.

SONET aggregation is proprietary to the JUNOS software and might not work with other software.

For aggregated Ethernet interfaces, you must include the vlan-tagging statement at the [edit interfaces ae*x*] hierarchy level to complete the association.

For more information, see “Configure Aggregated Ethernet Interfaces” on page 247 or “Configure Aggregated SONET/SDH Interfaces” on page 290.

Add an Interface Description to the Configuration

You can include a text description of each physical interface in the configuration file. Any descriptive text you include is displayed in the output of the show interfaces commands. It has no impact on the interface's configuration. To add a text description, include the description statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
description text;
```

The description can be a single line of text. If the text contains spaces, enclose it in quotation marks.

For information about describing logical units, see "Add a Logical Unit Description to the Configuration" on page 103.

Example: Add an Interface Description to the Configuration

Add a description to a SONET interface:

```
[edit interfaces so-1/1/0]
user@host# set description "BB: phl01 P12/0/0 - local wire"
[edit interfaces so-1/1/0]
user@host# commit
[edit interfaces so-1/1/0]
user@host# exit configuration-mode
cli> show interfaces so-1/1/0
so-1/1/0 {
    physical-interface index 9 snmp-ifindex 10;
    enabled physical-link up;
    description "BB: phl01 P12/0/0 - local wire";
    encapsulation cisco-hdlc;
...
```

Configure the Link Characteristics

By default, the router's management Ethernet interface, fxp0, autonegotiates whether to operate in full-duplex or half-duplex mode. Fast Ethernet interfaces can operate in either full-duplex or half-duplex mode, and all other interfaces can operate only in full-duplex mode. For Gigabit Ethernet, the link partner must also be set to full duplex.

To explicitly configure an Ethernet interface to operate in either full-duplex or half-duplex mode, include the link-mode statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
link-mode (full-duplex | half-duplex);
```

Configure the Media MTU

The default media MTU size used on a physical interface depends on the encapsulation being used on that interface. Table 3, Table 4, and Table 5 list the media MTU size by interface type and Table 6 lists the encapsulation overhead by encapsulation type.

Table 3: Media MTU Sizes by Interface Type for M5, M10, M20, and M40 Routers

Interface Type	Default Media MTU (Bytes)	Maximum MTU (Bytes)	Default IP Protocol MTU (Bytes)
ATM	4482	9192	4470
E1/T1	1504	9192	1500
E3/T3	4474	9192	4470
Fast Ethernet	1514		1500 (IPv4) 1497 (ISO)
4-port		9192	
8-port		1532	
Gigabit Ethernet	1514	9192	1500 (IPv4) 1497 (ISO)
SONET/SDH	4474	9192	4470

Table 4: Media MTU Sizes by Interface Type for M40e and M160 Routers

Interface Type	Default Media MTU (Bytes)	Maximum MTU (Bytes)	Default IP Protocol MTU (Bytes)
ATM	4482	9192	4470
E1/T1	1504	4500	1500
E3/T3	4474	4500	4470
Fast Ethernet	1514		1500 (IPv4) 1497 (ISO)
4-port		4500	
8-port		1532	
12-port (for M40e)		1532	
48-port		1532	
Gigabit Ethernet	1514		1500 (IPv4) 1497 (ISO)
1- or 2-port		9192	
4-port		4500	
10-gigabit (for M160)		9192	
SONET/SDH	4474		4470
4-port OC-3		4500	
4-port OC-3c		4500	
1-port OC-12		4500	
4-port OC-12		4500	
4-port OC-12c		4500	
1-port OC-48		4500	
2-port OC-3		9192	
2-port OC-3c		9192	
1-port OC-12c		9192	
1-port OC-48c		9192	
1-port OC-192		9192	
1-port OC-192c		9192	

Table 5: Media MTU Sizes by Interface Type for T-series Platforms

Interface Type	Default Media MTU (Bytes)	Maximum MTU (Bytes)	Default IP Protocol MTU (Bytes)
48-port Fast Ethernet	1514	1532	1500 (IPv4) 1497 (ISO)
Gigabit Ethernet	1514		1500 (IPv4) 1497 (ISO)
1- or 2-port		9192	
4-port		4500	
10-gigabit		9192	
SONET/SDH	4474	9192	4470


Note

The physical MTU for Ethernet interfaces does not include the 4-byte FCS field of the Ethernet frame.

A SONET interface operating in concatenated mode has a “c” added to the rate descriptor. For example, a concatenated OC-48 interface is referred to as OC-48c.

Table 6: Encapsulation Overhead by Encapsulation Type

Interface Encapsulation	Encapsulation Overhead (Bytes)
ATM Cell Relay	4
ATM PVC	12
Cisco HDLC	4
Frame Relay	4
Point-to-Point Protocol	4
Ethernet over ATM	28
Ethernet Version 2	14
Ethernet 802.3	17
802.1Q/Ethernet Version 2	18
802.1Q/Ethernet 802.3	21
Ethernet CCC	4
Ethernet TCC	18
Ethernet SNAP	22
802.1Q/Ethernet SNAP	26
VLAN CCC	4
Extended VLAN CCC	4
Extended VLAN TCC	22

The default media MTU is calculated as follows:

Default media MTU = Default IP MTU + encapsulation overhead

When you are configuring point-to-point connections, the MTU sizes on both sides of the connections must be the same. Also, when you are configuring point-to-multipoint connections, all interfaces in the subnet must use the same MTU size.

The actual frames transmitted also contain cyclic redundancy check (CRC) bits, which are not part of the media MTU. For example, the media MTU for a gigabit Ethernet interface is specified as 1500 bytes, but the largest possible frame size is actually 1504 bytes; you need to consider the extra bits in calculations of MTUs for interoperability.

For information about configuring the encapsulation on an interface, see “Configure Interface Encapsulation” on page 44.

To modify the default media MTU size for a physical interface, include the mtu statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
mtu bytes;
```

If you change the size of the media MTU, you must ensure that the size is equal to or greater than the sum of the protocol MTU and the encapsulation overhead. You configure the protocol MTU by including the mtu statement at the [edit interfaces *interface-name* unit *logical-unit-number* family *family*] hierarchy level, as discussed in “Set the Protocol MTU” on page 131.

Configure Interface Encapsulation

Point-to-Point Protocol (PPP) encapsulation is the default encapsulation type for physical interfaces. You need not configure encapsulation for any physical interfaces that support PPP encapsulation. If you do not configure encapsulation, PPP is used by default. For physical interfaces that do not support PPP encapsulation, you must configure an encapsulation to use for packets transmitted on the interface.

You can optionally configure an encapsulation on a logical interface, which is the encapsulation used within certain packet types. For more information about logical interface encapsulation, see “Configure the Encapsulation on a Logical Interface” on page 106.

Configure the Encapsulation on a Physical Interface

The physical interface encapsulation can be one of the following:

Point-to-Point Protocol (PPP)—Defined in RFC 1331, *The Point-to-Point Protocol (PPP) for the Transmission of Multiprotocol Datagrams over Point-to-Point Links*. PPP is the default encapsulation type for physical interfaces. E1, E3, SONET, T1, and T3 interfaces can use PPP encapsulation. Two related versions are supported:

Circuit cross-connect (CCC) version (ppp-ccc)—The logical interfaces do not require an encapsulation statement. When you use this encapsulation, you can configure the family ccc only.

Translational cross-connect (TCC) version (ppp-tcc)—Similar to CCC and has the same configuration restrictions, but used for circuits with different media on either side of the connection.

ATM Cell Relay—Connects two remote virtual circuits or ATM physical interfaces with a label-switched path (LSP). Traffic on the circuit is ATM cells.

You can configure an ATM PIC to use cell-relay accumulation mode. In this mode, the incoming cells (1 to 8 cells) are packaged into a single packet and forwarded to the label-switched path (LSP). For more information, see the *JUNOS Internet Software Configuration Guide: Getting Started*.

ATM PVC—Defined in RFC 1483, *Multiprotocol Encapsulation over ATM Adaptation Layer 5*. When you configure physical ATM interfaces with ATM PVC encapsulation, you can configure the logical interfaces with any of the ATM encapsulations listed in "Configure PPP Challenge Handshake Authentication Protocol" on page 47.

Ethernet over ATM—Allows ATM interfaces to connect to devices that support only bridged-mode protocol data units (PDUs).

Cisco HDLC—E1, E3, SONET, T1, and T3 interfaces can use Cisco HDLC encapsulation. Two related versions are supported:

CCC version (cisco-hdlc-ccc)—The logical interfaces do not require an encapsulation statement. When you use this encapsulation, you can configure the family ccc only.

TCC version (cisco-hdlc-tcc)—Similar to CCC and has the same configuration restrictions, but used for circuits with different media on either side of the connection.

Ethernet Cross-Connect—Ethernet interfaces without VLAN tagging can use Ethernet CCC encapsulation. Two related versions are supported:

CCC version (ethernet-ccc)—Ethernet interfaces with standard Tag Protocol ID (TPID) tagging can use Ethernet CCC encapsulation. One-port Gigabit Ethernet, two-port Gigabit Ethernet, four-port Gigabit Ethernet, and four-port Fast Ethernet PICs can use Ethernet CCC encapsulation. When you use this encapsulation, you can configure the family ccc only.

TCC version (ethernet-tcc)—Similar to CCC, but used for circuits with different media on either side of the connection. One-port Gigabit Ethernet, two-port Gigabit Ethernet, four-port Gigabit Ethernet, and four-port Fast Ethernet PICs can use Ethernet TCC encapsulation. Ethernet TCC is currently not supported on the T-series platforms.

Frame Relay—Defined in RFC 1490, *Multiprotocol Interconnect over Frame Relay*. E1, E3, SONET, T1, and T3 interfaces can use Frame Relay encapsulation. Two related versions are supported:

CCC version (frame-relay-ccc)—The same as standard Frame Relay for DLCIs 0 through 511. DLCIs 512 through 1022 are dedicated to CCC, and the logical interface must also have frame-relay-ccc encapsulation. When you use this encapsulation, you can configure the family ccc only.

TCC version (frame-relay-tcc)—Similar to Frame Relay CCC and has the same configuration restrictions, but is used for circuits with different media on either side of the connection.

VLAN Circuit Cross-Connect (CCC)—Ethernet interfaces with Virtual Local Area Network (VLAN) tagging enabled can use VLAN CCC encapsulation. When you use this encapsulation, you can configure the family ccc only.

Extended VLAN Cross-Connect—Gigabit Ethernet interfaces with Virtual Local Area Network (VLAN) tagging enabled can use extended VLAN cross-connect encapsulation. Two related versions of extended VLAN cross-connect are supported:

CCC version (extended-vlan-ccc)—Ethernet interfaces with 802.1Q tagging can use extended VLAN CCC encapsulation. (Ethernet interfaces with standard Tag Protocol ID (TPID) tagging can use Ethernet CCC encapsulation.) Extended VLAN CCC is not supported on four-port Gigabit Ethernet PICs. When you use this encapsulation, you can configure the family ccc only.

TCC version (extended-vlan-tcc)—Similar to CCC, but used for circuits with different media on either side of the connection. One-port Gigabit Ethernet, two-port Gigabit Ethernet, four-port Gigabit Ethernet, and four-port Fast Ethernet PICs can use Extended Ethernet TCC encapsulation. Extended Ethernet TCC is currently not supported on the T-series platforms.

To configure the encapsulation on a physical interface, include the encapsulation statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
encapsulation (atm-ccc-cell-relay | atm-pvc | cisco-hdlc | cisco-hdlc-ccc | cisco-hdlc-tcc | ethernet-ccc
| ethernet-tcc | ethernet-over-atm | frame-relay | frame-relay-ccc | frame-relay-tcc | ppp | ppp-ccc |
ppp-tcc | vlan-ccc | extended-vlan-ccc | extended-vlan-tcc);
```

When you configure a point-to-point encapsulation (such as PPP or Cisco HDLC) on a physical interface, the physical interface can have only one logical interface (that is, only one unit statement) associated with it. When you configure a multipoint encapsulation (such as Frame Relay), the physical interface can have multiple logical units, and the units can be either point-to-point or multipoint.

Ethernet interfaces in VLAN mode can have multiple logical interfaces. For encapsulation type vlan-ccc, VLAN IDs from 0 through 511 are reserved for normal VLANs, and VLAN IDs from 512 through 1023 are reserved for CCC VLANs. For encapsulation type extended-vlan-ccc, all VLAN IDs from 0 through 4094 are valid. For more information, see “Configure 802.1Q VLANs” on page 234. Ethernet CCC encapsulation for Ethernet interfaces with standard Tag Protocol ID (TPID) tagging requires that the physical interface have only a single logical interface.

When you configure a TCC encapsulation, some modifications are needed to handle VPN connections over unlike Layer 2 and Layer 2.5 links and terminate the Layer 2 and Layer 2.5 protocol locally. The router performs the following media-specific changes:

PPP TCC—Both Link Control Protocol (LCP) and Network Control Protocol (NCP) are terminated on the router. Internet Protocol Control Protocol (IPCP) IP address negotiation is not supported. The JUNOS software strips all PPP encapsulation data from incoming frames before forwarding them. For output, the next hop is changed to PPP encapsulation.

Cisco HDLC TCC—Keepalive processing is terminated on the router. The JUNOS software strips all Cisco HDLC encapsulation data from incoming frames before forwarding them. For output, the next hop is changed to Cisco HDLC encapsulation.

Frame Relay TCC—All Local Management Interface (LMI) processing is terminated on the router. The JUNOS software strips all Frame Relay encapsulation data from incoming frames before forwarding them. For output, the next hop is changed to Frame Relay encapsulation.

ATM—Operation, Administration, and Maintenance (OAM) and Interim Local Management Interface (ILMI) processing is terminated at the router. Cell relay is not supported. The JUNOS software strips all ATM encapsulation data from incoming frames before forwarding them. For output, the next hop is changed to ATM encapsulation.

Example: Configure the Encapsulation on a Physical Interface

Configure PPP encapsulation on a SONET interface. The second and third family statements allow IS-IS and MPLS to run on the interface:

```
[edit interfaces]
so-7/0/0 {
    encapsulation ppp;
    unit 0 {
        point-to-point;
        family inet {
            address 192.168.1.113/32 {
                destination 192.168.1.114;
            }
        }
        family iso;
        family mpls;
    }
}
```

Configure PPP Challenge Handshake Authentication Protocol

You can configure interfaces to support PPP Challenge Handshake Authentication Protocol (CHAP), as defined in RFC 1994. When CHAP is enabled, an interface with PPP encapsulation can authenticate its peer and can be authenticated by its peer.

By default, PPP CHAP is disabled. If CHAP is not explicitly enabled, the interface makes no CHAP challenges and denies all incoming CHAP challenges. To enable CHAP on links with PPP encapsulation, you must create a global mapping of link names and authentication data associated with those links, and you must create a per-interface configuration.

To create a global mapping of link names and authentication data, you configure access profiles using statements in the access hierarchy; for more information about configuring access profiles, see the *JUNOS Internet Softw are Configur ation Guide: Getting St arted*. The per-interface configuration includes a reference to an access profile. When a specified interface receives CHAP challenges and responses, the named access profile in the packet is used to look up the shared secret, as defined in RFC 1994.

To configure PPP CHAP on an interface with PPP encapsulation, include the chap statement at the [edit interfaces *interface-name* ppp-options] hierarchy level:

```
[edit interfaces interface-name ppp-options]
chap {
    access-profile name;
    local-name name;
    passive;
}
```

On each interface with PPP encapsulation, you can configure the following PPP CHAP properties:

Assign an Access Profile to an Interface on page 48

Configure the Local Name on page 48

Configure Passive Mode on page 48

Assign an Access Profile to an Interface

The CHAP authentication method depends upon a “secret” known only to the authenticator and that peer. The secret is not sent over the link. An access profile is a map between peer names (or “clients”) and the secrets associated with their respective links.

When an interface receives CHAP challenges and responses, the value of the access profile is extracted from the packets. This value is the identity of the peer for a specified interface. For information about configuring access profiles and secrets, see the *JUNOS Internet Softw are Configur ation Guide: Getting St arted*.

To assign an access profile to an interface, include the access-profile statement at the [edit interfaces *interface-name* ppp-options chap] hierarchy level:

```
[edit interfaces interface-name ppp-options chap]
access-profile name;
```

access-profile is a mandatory statement of the chap hierarchy. If an interface receives a CHAP challenge or response with a value for *name* that is not in the named access profile, the link is immediately dropped.

Configure the Local Name

You can configure the value sent in CHAP challenge and response packets on a per-interface basis. By default, each interface uses the router's system host name as the name sent in CHAP challenge and response packets.

To configure the name sent in CHAP challenge and response packets, include the local-name statement at the [edit interfaces *interface-name* ppp-options chap] hierarchy level:

```
[edit interfaces interface-name ppp-options chap]
local-name name;
```

Configure Passive Mode

By default, when the chap statement is present, the interface always challenges its peer and responds to challenges from its peer. You can configure the interface not to challenge its peer, and only respond when challenged. To configure the interface not to challenge its peer, include the passive statement at the [edit interfaces *interface-name* ppp-options chap] hierarchy level:

```
[edit interfaces interface-name ppp-options chap]
passive;
```

Example: Configure CHAP

Configure CHAP:

```
[edit access]
    profile pe-A-ppp-clients;
        client cpe-1 chap-secret "$1$dQYsZ$B5ojUeUjDsUo.yKwcCZ0"; # SECRET-DATA
        client cpe-2 chap-secret "$1$kdAsfaDAfkdjDsASxfafdKdFKJ"; # SECRET-DATA
    }
}

[edit interfaces so-1/2/0]
encapsulation ppp;
ppp-options {
    chap {
        access-profile pe-A-ppp-clients;
        local-name "pe-A-so-1/1/1";
    }
}

[edit interfaces so-1/1/2]
encapsulation ppp;
ppp-options {
    chap {
        access-profile pe-A-ppp-clients;
        local-name “pe-A-so-1/1/2";
    }
}
```

Configure the Interface Speed

By default, the router's management Ethernet interface, fxp0, autonegotiates whether to operate at 10 Mbps or 100 Mbps. All other interfaces automatically choose the correct speed based on the PIC type and whether the PIC is configured to operate in multiplexed mode (using the no-concatenate statement in the [edit chassis] configuration hierarchy, as described in "Configure Channelized Interfaces" on page 191).

To configure the management Ethernet interface to operate at 10 Mbps or 100 Mbps, include the speed statement at the [edit interfaces fxp0] hierarchy level:

```
[edit interfaces fxp0]
speed (10m | 100m);
```

Configure the MAC Address on the Management Ethernet Interface

By default, the router's management Ethernet interface (fxp0) uses as its MAC address the MAC address that is burned into the Ethernet card. To display this address, enter the show interface fxp0 operational mode command.

To change the management Ethernet interface's MAC address, include the loop-timing statement at the [edit interfaces fxp0] hierarchy level:

```
[edit interfaces fxp0]
mac mac-address;
```

Specify the MAC address as six hexadecimal bytes in one of the following formats: *nnnn.nnnn.nnnn* (for example, 0011.2233.4455) or *nn:nn:nn:nn:nn:nn* (for example, 00:11:22:33:44:55).

Configure Keepalives

By default, physical interfaces configured with Cisco HDLC or PPP encapsulation send keepalive packets at 10-second intervals. The Frame Relay term for keepalives is Local Management Interface (LMI) packets; the JUNOS software supports both ANSI T1.617 Annex D LMIs and ITU Q933 Annex A LMIs. On ATM networks, Operation, Administration, and Maintenance (OAM) cells perform the same function. You configure OAM cells at the logical interface level; for more information, see "Define the ATM OAM F5 Loopback Cell Period" on page 183.

To disable the sending of keepalives on a physical interface, include the no-keepalives statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
no-keepalives;
```

For back-to-back Frame Relay connections, either disable the sending of keepalives on both sides of the connection, or configure one side of the connection as data terminal equipment (DTE) (the default JUNOS configuration) and the other as data circuit-terminating equipment (DCE).

To explicitly enable the sending of keepalives on a physical interface, include the keepalives statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
keepalives;
```

On interfaces configured with Cisco HDLC or PPP encapsulation, you can configure the following three keepalive parameters; note that Frame Relay encapsulation is not affected by these options:

interval *seconds*—The time in seconds between successive keepalive requests. The range is 1 second through 32767 seconds, with a default of 10 seconds.

down-count *number*—The number of keepalive packets a destination must fail to receive before the network takes a link down. The range is 1 through 255, with a default of 3.

up-count *number*—The number of keepalive packets a destination must receive to change a link's status from down to up. The range is 1 through 255, with a default of 1.

To change one or more of the default keepalive values, include the appropriate option at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
keepalives <interval seconds> <down-count number> <up-count number>;
```

For interfaces using multipoint or multicast connections over Frame Relay encapsulation, if keepalives are enabled, the number of possible DLCI configurations is limited by the MTU size selected for the interface. To calculate the available DLCIs, use the formula (MTU – 12) / 5. To increase the number of possible DLCIs, disable keepalives on the interface.

Configure Keepalive Settings on Frame Relay LMI

On interfaces configured with Frame Relay connections, you can tune the keepalive settings by using the lmi statement. A Frame Relay interface can be either a DCE or DTE (the default JUNOS configuration). DTE acts as a master, requesting status from the DCE part of the link.

By default, the JUNOS software uses ANSI Annex D LMIs. To use ITU Annex A LMIs instead, include the lmi-type itu statement at the [edit interfaces *interface-name* lmi] hierarchy level:

```
[edit interfaces interface-name lmi]
lmi-type itu;
```

To configure Frame Relay keepalive parameters, include the lmi statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
lmi {
    lmi-type (ansi | itu);
    n391dte number;
    n392dce number;
    n392dte number;
    n393dce number;
    n393dte number;
    t391dte seconds;
    t392dce seconds;
}
```

You can set the following parameters:

n391dte—DTE full status polling interval. The DTE sends a status inquiry to the DCE at the interval specified by t391dte. n391dte specifies the frequency at which these inquiries expect a full status report; for example, a n391dte value of 10 would specify a full status report in response to every tenth inquiry. The intermediate inquiries ask for a keepalive exchange only. The range is 1 through 255, with a default value of 6.

n392dce—DCE error threshold. The number of errors required to bring down the link, within the event-count specified by n393dce. The range is 1 through 10, with a default value of 3.

n392dte—DTE error threshold. The number of errors required to bring down the link, within the event-count specified by n393dte. The range is 1 through 10, with a default value of 3.

n393dce—DCE monitored event-count. The range is 1 through 10, with a default value of 4.

n393dte—DTE monitored event-count. The range is 1 through 10, with a default value of 4.

t391dte—DTE keepalive timer. Period at which the DTE sends out a keepalive response request to the DCE and updates status depending on the error threshold value. The range is 5 through 30 seconds, with a default value of 10 seconds.

t392dce—DCE keepalive timer. Period at which the DCE checks for keepalive responses from the DTE and updates status depending on the DCE error threshold value. The range is 5 through 30 seconds, with a default value of 15 seconds.

Configure the Clock Source

For interfaces such as SONET that can use different clock sources, you can configure the source of the transmit clock on each interface. The source can be internal (also called line timing or normal timing) or external (also called loop timing). The default source is internal, which means that each interface uses the router's internal stratum 3 clock.

For DS-3 channels on a Channelized OC-12 interface, the clocking statement is supported only for channel 0; it is ignored if included in the configuration of channels 1 through 11. The clock source configured for channel 0 applies to all channels on the Channelized OC-12 interface. The individual DS-3 channels use a gapped 45-MHz clock as the transmit clock.

On Channelized STM-1 interfaces, you should configure the clock source at one side of the connection to be internal (the default JUNOS configuration) and configure the other side of the connection to be external.

To configure loop timing on an interface, include the clocking external statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
clocking external;
```

To explicitly configure line timing on an interface, include the clocking internal statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
clocking internal;
```

Configure the Router as a DCE

By default, when you configure an interface with Frame Relay encapsulation, the router is assumed to be data terminal equipment (DTE). That is, the router is assumed to be at a terminal point on the network. To configure the router to be data circuit-terminating equipment (DCE), include the dce statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
dce;
```

When you configure the router to be a DCE, keepalives are disabled by default.

For back-to-back Frame Relay connections, either disable the sending of keepalives on both sides of the connection, or configure one side of the connection as a DTE (the default JUNOS configuration) and the other as a DCE.

Configure Receive and Transmit Leaky Bucket Properties

Congestion control is particularly difficult in high-speed networks with high volumes of traffic. When congestion occurs in such a network, it is usually too late to react. You can avoid congestion by regulating the flow of packets into your network. Smoother flows prevent bursts of packets from arriving at (or being transmitted from) the same interface and causing congestion.

For all interface types except ATM, Fast Ethernet, and Gigabit Ethernet, you can configure leaky bucket properties, which allow you to limit the amount of traffic received on and transmitted by a particular interface. You effectively specify what percentage of the interface's total capacity can be used to receive or transmit packets. You might want to set leaky bucket properties to limit the traffic flow from a link that is known to transmit high volumes of traffic.

Instead of configuring leaky bucket properties, you can limit traffic flow by configuring policers. Policers work on all interfaces. For more information, see "Apply Policers" on page 141 and the *JUNOS Internet Software Configuration Guide: Policy Framework*.

The leaky bucket is used at the host-network interface to allow packets into the network at a constant rate. Packets might be generated in a bursty manner, but after they pass through the leaky bucket, they enter the network evenly spaced. In some cases, you might want to allow short bursts of packets to enter the network without smoothing them out. By controlling the number of packets that can accumulate in the bucket, the threshold property controls burstiness. The maximum number of packets entering the network in *t* time units is threshold + rate * *t*.

By default, leaky buckets are disabled, and the interface can receive and transmit packets at the maximum line rate.

For each DS-3 channel on a Channelized OC-12 interface, you can configure unique receive and transmit buckets. To configure leaky bucket properties, include one or both of the receive-bucket and transmit-bucket statements at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
receive-bucket {
    overflow (tag | discard);
    rate percentage;
    threshold number;
}
transmit-bucket {
    overflow (discard);
    rate percentage;
    threshold number;
}
```

In the rate option, specify the percentage of the interface line rate that is available to receive or transmit packets. The percentage can be a value from 0 (none of the interface line rate is available) to 100 (the maximum interface line rate is available). For example, when you set the line rate to 33, the interface receives or transmits at one third of the maximum line rate.

In the threshold option, specify the bucket threshold, which controls the burstiness of the leaky bucket mechanism. The larger the value, the more bursty the traffic, which means that over a very short amount of time the interface can receive or transmit close to line rate, but the average over a longer time is at the configured bucket rate. The threshold can be a value from 0 through 16777215 bytes. For ease of entry, you can enter *number* either as a complete decimal number or as a decimal number followed by the abbreviation k (1,000) or m (1,000,000). For example, the entry threshold 2m corresponds to a threshold of 2,000,000 bytes.

In the overflow option, specify how to handle packets that exceed the threshold:

tag—(receive-bucket only) Tag, count, and process received packets that exceed the threshold.

discard—Discard received packets that exceed the threshold. No counting is done.

Configure Accounting

Juniper Networks routers can collect various kinds of data about traffic passing through the router. You can set up one or more *accounting profiles* that specify some common characteristics of this data, including the following:

The fields used in the accounting records

The number of files that the router retains before discarding, and the number of bytes per file

The polling period that the system uses to record the data

You configure the profiles using statements in the accounting-options hierarchy; for more information, see the *JUNOS Internet Software Configuration Guide: Getting Started*. You must assign a unique name for each accounting profile; this name cross-references the information specified in the accounting-options hierarchy with interfaces or firewall configuration statements.

Configure Physical Interface Profiles

There are two types of accounting profiles: interface profiles and filter profiles. They have different configuration statements in the accounting-profiles hierarchy, and are implemented separately in either the interfaces or firewall hierarchy. If you reference the same profile from both a firewall filter and an interface statement within the same configuration, it causes an error.

The following is a sample accounting-options profile for an interface; for more information, see the *JUNOS Internet Software Configuration Guide: Network Management*.

```
[edit]
accounting-options {
    file if_stats {
        size 4m files 10 transfer-interval 15;
        archive-sites {
            "ftp://login:password@host/path";
        }
    }
    interface-profile if_profile {
        interval 15;
        file if_stats {
            fields {
                input-bytes;
                output-bytes;
                input-packets;
                output-packets;
                input-errors;
                output-errors;
            }
        }
    }
}
```

To enable accounting on an interface, include the accounting-profile statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
accounting-profile name;
```

You can also reference profiles by logical unit; for more information about referencing profiles by logical unit, see “Configure the Logical Interface Profile” on page 104. For information about configuring a firewall filter accounting profile, see the *JUNOS Internet Software Configuration Guide: Policy Framework*.

Configure BERT Properties

You can configure any of the following interfaces to execute a bit error rate test (BERT) when the interface receives a request to run this test: E1, E3, T1, T3, and the channelized DS-3, OC-3, OC-12, and STM-1 interfaces. On all of the specified interface types, you set the duration of the test and the error rate to include in the bit stream by including the bert-period and bert-error-rate statements at the [edit interfaces *interface-name interface*-options] hierarchy level:

```
[edit interfaces interface-name interface-options]
bert-error-rate rate;
bert-period seconds;
```

seconds is the duration of the BERT procedure, in seconds. The test can last from 1 to 240 seconds; the default is 10 seconds.

rate is the bit error rate. This can be an integer in the range 0 through 7, which corresponds to a bit error rate in the range 10^{-0} (that is, 1 error per bit) to 10^{-7} (that is, 1 error per 10 million bits).

algorithm is the pattern to send in the bit stream. The algorithm for the E1 BERT procedure is pseudo-2e15-o151 (pattern is 2^{15}–1, as defined in the CCITT/ITU O.151 standard). On T1, E3, and T3 interfaces, you can also select the pattern to send in the bit stream by including the bert-algorithm statement at the [edit interfaces *interface-name interface*-options] hierarchy level:

```
[edit interfaces interface-name interface-options]
bert-algorithm algorithm;
```

For a list of supported algorithms, see the CLI possible completions, for example:

```
[edit interfaces t1-0/0/0 t1-options]
user@host# set bert-algorithm ?
Possible completions:
pseudo-2e11-o152    Pattern is 2^11 -1 (per O.152 standard)
pseudo-2e15-o151    Pattern is 2^15 - 1 (per O.152 standard)
pseudo-2e20-o151    Pattern is 2^20 - 1 (per O.151 standard)
pseudo-2e20-o153    Pattern is 2^20 - 1 (per O.153 standard)
```

See individual interface types for specific hierarchy information. For information about running the BERT procedure, see the *JUNOS Internet Software Operational Mode Command Reference*.

Table 7 shows the BERT capabilities for various interface types.

Table 7: BERT Capabilities by Interface Type

Interface	T1 BERT	T3 BERT	Comments
E1 or T1	Yes (port 0–3)	Yes (port 0–3)	Single port at a time Limited algorithms
E3 or T3	Yes (port 0–3)	Yes (port 0–3)	Single port at a time
Channelized OC-12	N/A	Yes (channel 0–11)	Single channel at a time Limited algorithms No bit count
Channelized STM-1	Yes (channel 0–62)	N/A	Multiple channels Only one algorithm No error insert No bit count
Channelized T3 and Multichannel T3	Yes (channel 0–27)	Yes (port 0–3 on channel 0)	Multiple ports and channels Limited algorithms for T1 No error insert for T1 No bit count for T1

Trace Operations of an Individual Router Interface

To trace the operations of individual router interfaces, include the traceoptions statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
traceoptions {
    flag flag <disable>;
}
```

You can specify the following interface tracing flags:

all—Trace all interface operations.

event—Trace all interface events.

ipc—Trace all interface IPC messages.

media—Trace all interface media changes.

The interfaces traceoptions statement does not support a trace file. The logging is done by the kernel, so the tracing information is placed in the system syslog files.

For more informations about trace operations, see “Trace Interface Operations” on page 171.

Damp Interface Transitions

By default, when an interface changes from being up to being down, or from down to up, this transition is advertised immediately to the router software and hardware. In some situations, for example, when an interface is connected to an ADM or WDM, or to protect against SONET framer holes, you might want to damp interface transitions, thereby not advertising the interface’s transition until a certain period of time has passed, called the *hold-time*. When you have damped interface transitions and the interface goes from up to down, the interface is not advertised to the rest of the system as being down until it has remained down for the hold-time period. Similarly when an interface goes from down to up, it is not advertised as being up until it has remained up for the hold-time period.

To damp interface transitions, include the hold-time statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
hold-time up milliseconds down milliseconds;
```

The time can be a value from 0 through 65,534 milliseconds. Upon execution, the time value that you specify is rounded up to the nearest whole second; therefore, we recommend that you configure the up and down options to multiples of 1000. The default value is 0, which means that interface transitions are not damped.

Disable SNMP Notifications on Physical Devices

By default, SNMP notifications are sent when the interface or connection state changes. To disable this notification on the physical interface, include the no-traps statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
no-traps;
```

Disable a Physical Interface

You can disable a physical interface, marking it as being down, without removing the interface configuration statements from the configuration. To do this, include the disable statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
disable;
```

Example: Disable a Physical Interface

Disable a physical interface:

```
[edit interfaces]
so-1/1/0 {
    mtu 8000;
    clocking internal;
    encapsulation ppp;
    sonet-options {
        fcs 16;
    }
    unit 0 {
        family inet {
            address 12.12.12.21/32 {
                destination 12.12.12.22;
            }
        }
    }
}
[edit interfaces]
user@host# set so-1/1/0 disable
[edit interfaces]
user@host# show so-1/1/0
so-1/1/0 {
    disable;          # Interface is marked as disabled
    mtu 8000;
    clocking internal;
    encapsulation ppp;
    sonet-options {
        fcs 16;
    }
    unit 0 {
        family inet {
            address 12.12.12.21/32 {
                destination 12.12.12.22;
            }
        }
    }
}
```

Configure Aggregated Ethernet Interfaces

Link aggregation of Ethernet interfaces is defined in the IEEE 802.3ad standard. The JUNOS implementation of 802.3AD balances traffic across the member links within an aggregated Ethernet bundle based on the Layer 3 information carried in the packet. This implementation uses the same load balancing algorithm used for per-packet load balancing. For information about per-packet load balancing, see the *JUNOS Internet Software Guide: Routing and Routing Protocols*.

The JUNOS software does not provide load balancing for multicast traffic on aggregated interfaces. If a link carrying multicast data goes down, another link carries the traffic. This provides redundancy, not more bandwidth.

You configure an aggregated Ethernet virtual link by specifying the link number as a physical device and then associating a set of ports that have the same speed and are in full-duplex mode. The physical interfaces can be either Fast Ethernet or Gigabit Ethernet devices, but must not intermix within the same aggregated link.

To specify aggregated Ethernet interfaces, include the vlan-tagging statement at the [edit interfaces ae*x*] hierarchy level and include the vlan-id statement at the [edit interfaces ae*x* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces]
aex {
    vlan-tagging;
    unit logical-unit-number {
        vlan-id number;
        family inet {
            address address;
        }
    }
}
```

By default, no aggregated Ethernet interfaces are created. You must define the number of aggregated Ethernet interfaces by including the device-count statement at the [edit chassis aggregated-devices ethernet] hierarchy level:

```
[edit chassis]
aggregated-devices {
    ethernet {
        device-count number;
    }
}
```

The maximum number of aggregated interfaces is 16, and the assigned number can range from 0 through 15. For information about configuring aggregated devices, see the *JUNOS Internet Software Guide: Getting Started*.

You must also specify the constituent physical links by including the 802.3ad statement at the [edit interfaces *interface-name* fastether-options] or [edit interfaces *interface-name* gigether-options] hierarchy level; for more information, see “Configure Ethernet Link Aggregation” on page 78. You can optionally specify other physical properties that apply specifically to the aggregated Ethernet interfaces; for details, see “Configure Ethernet Physical Interface Properties” on page 77. For a sample configuration, see “Configure Aggregated Ethernet Interfaces” on page 247.

To delete an aggregated Ethernet interface from the configuration, issue the delete interfaces ae*x* command at the [edit] hierarchy level in configuration mode:

```
[edit]
user@host# delete interfaces aex
```

If you delete an aggregated Ethernet interface from the configuration, the software removes the configuration statements related to ae*x* and sets this interface to down state. However, the aggregated Ethernet interface is not deleted until you delete the chassis aggregated-devices ethernet device-count configuration statement.

Configure ATM Physical Interface Properties

For ATM physical interfaces, you can configure two ATM-specific physical device properties: the maximum number of virtual circuits (VCs) allowed on a virtual path (VP) and communication with directly attached ATM switches. You configure these properties by including the atm-options statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
atm-options {
    vpi vpi-identifier maximum-vcs maximum-vcs;
    promiscuous-mode;
    ilmi;
}
```

You can configure the following ATM-specific properties:

Configure the Maximum Number of Virtual Circuits on a Virtual Path on page 60

Configure ATM Cell-Relay Promiscuous Mode on page 61

Configure Communication with Directly Attached ATM Switches on page 63

Configure the Maximum Number of Virtual Circuits on a Virtual Path

You configure the maximum number of virtual circuits allowed on a virtual path so that sufficient memory on the ATM PIC can be allocated for each VC. When configuring ATM interfaces on the router, you must include this statement.

To configure the largest numbered VCs on a VP, include the vpi statement at the [edit interfaces *interface-name* atm-options] hierarchy level:

```
[edit interfaces interface-name atm-options]
vpi vpi-identifier maximum-vcs maximum-vcs;
```

The VP identifier can be a value from 0 through 255. For most interfaces, you can define a maximum of 4090 VCs per interface. The highest numbered VC value you can configure is 4089. For ATM OC-3 interfaces, you can define a maximum of 8186 VCs per interface. For ATM OC-12 interfaces, you can define a maximum of 16,378 VCs per interface. Promiscuous mode removes these limits. For more information, see “Configure ATM Cell-Relay Promiscuous Mode” on page 61.

All the VPIs that you configure in the atm-options statement are stored by the software in a single table. If you modify the VPIs, for example, by editing them in configuration mode or by issuing a load override command, all VCs on the interface are closed and then reopened, resulting in a temporary loss of connectivity for all VCs on the interface.

You can also include some of the statements in the sonet-options statement to set SONET/SDH parameters on ATM interfaces as described in "Configure SONET/SDH Physical Interface Properties" on page 82.

Configure ATM Cell-Relay Promiscuous Mode

For ATM interfaces with atm-ccc-cell-relay encapsulation, you can map all incoming cells from either an interface port or a virtual path (VP) to a single LSP without restricting the VCI number. Promiscuous mode allows you to map traffic from all 65,535 VCIs to a single LSP, or from all 256 VPIs to a single LSP.

To map incoming traffic from a port or VC to an LSP, include the promiscuous-mode statement at the [edit interfaces *interface-name* atm-options] hierarchy level:

```
[edit interfaces interface-name]
atm-options {
    promiscuous-mode;
}
```

For multiport PICs, all ports must be configured in either promiscuous mode or non-promiscuous mode. For promiscuous mode, you must configure all ports with atm-ccc-cell-relay encapsulation. For multiport ATM PICs such as OC-3 or T3, configuring one port or interface in promiscuous mode places all the ports or interfaces on that PIC in promiscuous mode.

When you configure interfaces to use promiscuous mode, you cannot configure VCIs.

To map incoming traffic from a port to an LSP, you must include the allow_any_vci statement at the [edit interfaces *interface-name* unit 0] hierarchy level, as shown in the following example:

```
[edit interfaces at-1/2/0]
encapsulation atm-ccc-cell-relay;
atm-options {
    promiscuous-mode;
}
unit 0 {
    allow_any_vci;
}
```

When you include the allow_any_vci statement, you cannot configure other logical interfaces in the same physical interface.

Next, you must map unit 0 to an LSP using the CCC connection, as shown in the following example:

```
protocols {
    connections {
        remote-interface-switch router-a-router-c {
            interface at-1/2/0.0;
            transmit-lsp lsp1
            receive-lsp lsp2;
        }
    }
}
```

To map a VPI to an LSP, you must define the allowed VPIs. You can configure one or more logical interfaces, each mapped to a different VPI. You can then route traffic from each of these interfaces to different LSPs, as shown in the following example:

```
[edit interfaces at-1/1/0]
encapsulation atm-ccc-cell-relay;
atm-options {
    promiscuous-mode {
        vpi 10;
        vpi 20;
    }
}
unit 0 {
    vpi 10;
unit 1 {
    vpi 20;

[edit interfaces at-3/1/0]
encapsulation atm-ccc-cell-relay;
atm-options {
    promiscuous-mode;
        vpi 10
        vpi 20;
    }
}
unit 0 {
    vpi 10;
unit 1 {
    vpi 20;

protocols {
    mpls {
protocols {
    connections {
        interface-switch router-a-router-c {
            interface at-1/1/0.0;
            interface at-3/1/0.0;
        }
        interface-switch router-a-router-c {
            interface at-1/1/0.1;
            interface at-3/1/0.1;
        }
    }
}
```

Configure Communication with Directly Attached ATM Switches

You configure communication to directly attached ATM switches to enable querying of the IP addresses and port numbers of the switches. You query the switch by entering the following command:

```
user@host> show ilmi interface interface-name
```

The router uses VC 0.16 to communicate with the ATM switch.

To configure communication between the router and its directly attached ATM switches, include the ilmi statement at the [edit interfaces *interface-name* atm-options] hierarchy level:

```
[edit interfaces interface-name atm-options]
ilmi;
```

Configure Channelized OC-12 Interface Properties

To configure Channelized OC-12 interface properties, you can include the sonet-options and t3-options statements. Some of the SONET/SDH options are ignored and some can only be configured for channel 0, although they apply equally to all channels.

You can configure 12 channels per interface, and each interface can have logical interfaces, the same as other physical interfaces. The long-buildout statement under t3-options is also ignored. For more information, see “Configure SONET/SDH Physical Interface Properties” on page 82 and “Configure E3 and T3 Physical Interface Properties” on page 70. Table 8 summarizes the OC-12 to DS-3 numbering scheme.

Table 8: OC-12 to DS-3 Numbering Scheme

2-Level STS-1 Number (STS-3,STS-1)	1-Level STS Number	OC-12 to DS-3 PIC DS-3 Number
1,1	1	0
1,2	2	1
1,3	3	2
2,1	4	3
2,2	5	4
2,3	6	5
3,1	7	6
3,2	8	7
3,3	9	8
4,1	10	9
4,2	11	10
4,3	12	11

Configure E1 and T1 Physical Interface Properties

To configure E1-specific physical interface properties, include the e1-options statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
e1-options {
    bert-error-rate rate;
    bert-period seconds;
    fcs (32 | 16);
    framing (g704 | g704-no-crc4 | unframed);
    idle-cycle-flag (flags | ones);
    loopback (local | remote);
    start-end-flag (shared | filler);
    timeslots slot-number;
}
```

To configure T1-specific physical interface properties, include the t1-options statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
t1-options {
    bert-error-rate rate;
    bert-period seconds;
    buildout (0-133 | 133-266 | 266-399 | 399-532 | 532-655);
    byte-encoding (nx64 | nx56);
    fcs (32 | 16);
    framing (sf | esf);
    idle-cycle-flag (flags | ones);
    invert-data;
    line-encoding (ami | b8zs);
    loopback (local | remote);
    start-end-flag (shared | filler);
    timeslots slot-number;
}
```

You can configure the following E1-specific and T1-specific properties:

Configure E1 and T1 BERT Properties on page 65

Configure T1 Buildout on page 66

Configure T1 Byte Encoding on page 66

Configure E1 and T1 Data Inversion on page 66

Configure the E1 and T1 Frame Checksum on page 66

Configure E1 Framing on page 67

Configure T1 Framing on page 67

Configure the E1 and T1 Idle Cycle Flag on page 68

Configure T1 Line Encoding on page 68

Configure E1 and T1 Loopback Capability on page 68

Configure the E1 and T1 Start and End Flags on page 69

Configure the E1 and T1 Timeslots on page 69

See also the following properties, which apply to a number of different interfaces:

Configure the Media MTU on page 41

Configure the Encapsulation on a Physical Interface on page 44

Configure the Clock Source on page 52

Configure Receive and Transmit Leaky Bucket Properties on page 53

Configure E1 and T1 BERT Properties

You can configure an E1 or a T1 interface to execute a bit error rate test (BERT) when the interface receives a request to run this test. You specify the duration of the test and the error rate to include in the bit stream by including the bert-period and bert-error-rate statements at the [edit interfaces *interface-name* e1-options] or [edit interfaces *interface-name* t1-options] hierarchy level:

```
bert-error-rate rate;
bert-period seconds;
```

seconds is the duration of the BERT procedure, in seconds. The test can last from 1 to 240 seconds; the default is 10 seconds.

rate is the bit error rate. This can be an integer in the range 0 through 7, which corresponds to a bit error rate in the range 10^{-0} (that is, 1 error per bit) to 10^{-7} (that is, 1 error per 10 million bits).

algorithm is the pattern to send in the bit stream. The algorithm for the E1 BERT procedure is pseudo-2e15-o151 (pattern is 2^{15}–1, as defined in the CCITT/ITU O.151 standard).

On T1 interfaces, you can also select the pattern to send in the bit stream by including the bert-algorithm statement at the [edit interfaces *interface-name interface*-options] hierarchy level:

```
[edit interfaces interface-name interface-options]
bert-algorithm algorithm;
```

For a list of supported algorithms, see the CLI possible completions, for example:

```
[edit interfaces t1-0/0/0 t1-options]
user@host# set bert-algorithm ?
Possible completions:
pseudo-2e11-o152     Pattern is 2^11 -1 (per O.152 standard)
pseudo-2e15-o151     Pattern is 2^15 - 1 (per O.152 standard)
pseudo-2e20-o151     Pattern is 2^20 - 1 (per O.151 standard)
pseudo-2e20-o153     Pattern is 2^20 - 1 (per O.153 standard)
```

See individual interface types for specific hierarchy information. For information about running the BERT procedure, see the *JUNOS Internet Software Operational Mode Command Reference*.

Configure T1 Buildout

A T1 interface has five possible setting ranges for the T1 line buildout: 0-133, 133-266, 266-399, 399-532, or 532-655 feet. By default, the T1 interface uses the shortest setting (0-133).

To have the interface support one of the longer distance ranges, include the buildout statement with the appropriate value at the [edit interfaces *interface-name* t1-options] hierarchy level:

```
[edit interfaces interface-name t1-options]
buildout 532-655;
```

Configure T1 Byte Encoding

By default, T1 interfaces use a byte encoding of 8 bits per byte (nx64). You can configure an alternative byte encoding of 7 bits per byte (nx56).

To have the interface use 7 bits per byte encoding, include the byte-encoding statement at the [edit interfaces *interface-name* t1-options] hierarchy level, specifying the nx56 option:

```
[edit interfaces interface-name t1-options]
byte-encoding nx56;
```

To explicitly configure nx64 byte encoding, include the byte-encoding statement at the [edit interfaces *interface-name* t1-options] hierarchy level, specifying the nx64 option:

```
[edit interfaces interface-name t1-options]
byte-encoding nx64;
```

Configure E1 and T1 Data Inversion

By default, data inversion is disabled. To enable data inversion at the HDLC level, include the invert-data statement at the[edit interfaces *interface-name* e1-options] or [edit interfaces *interface-name* t1-options] hierarchy level:

```
invert-data;
```

When you enable data inversion, all data bits in the data stream are transmitted as inverted; that is, zeroes are transmitted as ones and ones as zeroes. Data inversion is normally used only in AMI mode to guarantee ones density in the transmitted stream.

Configure the E1 and T1 Frame Checksum

By default, E1 and T1 interfaces use a 16-bit frame checksum. You can configure a 32-bit checksum, which provides more reliable packet verification. However, some older equipment might not support 32-bit checksums.

To configure a 32-bit checksum, include the fcs 32 statement at the [edit interfaces *interface-name* e1-options] or [edit interfaces *interface-name* t1-options] hierarchy level:

```
fcs 32;
```

To return to the default 16-bit frame checksum, delete the fcs 32 statement from the configuration:

```
[edit]
user@host# delete interfaces t1-fpc/pic/port t1-options fcs 32
```

To explicitly configure a 16-bit checksum, include the fcs 16 statement at the [edit interfaces *interface-name* e1-options] or [edit interfaces *interface-name* t1-options] hierarchy level:

```
fcs 16;
```

Configure E1 Framing

By default, E1 interfaces use the G704 framing mode. You can configure the alternative unframed mode if needed.

To have the interface use the unframed mode, include the framing statement at the [edit interfaces *interface-name* e1-options] hierarchy level, specifying the unframed option:

```
[edit interfaces interface-name e1-options]
framing unframed;
```

To explicitly configure G704 framing, include the framing statement at the [edit interfaces *interface-name* e1-options] hierarchy level, specifying the g704 option:

```
[edit interfaces interface-name e1-options]
framing g704;
```

By default, G704 framing uses CRC4. To explicitly configure an interface's G704 framing to not use CRC4, include the framing statement at the [edit interfaces *interface-name* e1-options] hierarchy level, specifying the g704-no-crc4 option:

```
[edit interfaces interface-name e1-options]
framing g704-no-crc4;
```

Configure T1 Framing

By default, T1 interfaces use ESF (extended super frame) framing format. You can configure SF (super frame) format as an alternative.

To have the interface use the SF framing format, include the framing statement at the [edit interfaces *interface-name* t1-options] hierarchy level, specifying the sf option:

```
[edit interfaces interface-name t1-options]
framing sf;
```

To explicitly configure ESF framing, include the framing statement at the [edit interfaces *interface-name* t1-options] hierarchy level, specifying the esf option:

```
[edit interfaces interface-name t1-options]
framing esf;
```

Configure the E1 and T1 Idle Cycle Flag

By default, E1 and T1 interfaces transmit the value 0x7E in the idle cycles. To have the interface transmit the value 0xFF (all ones) instead, include the idle-cycle-flag statement at the [edit interfaces *interface-name* e1-options] or [edit interfaces *interface-name* t1-options] hierarchy level, specifying the ones option:

```
idle-cycle-flag ones;
```

To explicitly configure the default value of 0x7E, include the idle-cycle-flag statement with the flags option:

```
idle-cycle-flag flags;
```

Configure T1 Line Encoding

By default, T1 interfaces use B8ZS line encoding. You can configure AMI line encoding if necessary.

To have the interface use AMI line encoding, include the line-encoding statement at the [edit interfaces *interface-name* t1-options] hierarchy level, specifying the ami option:

```
[edit interfaces interface-name t1-options]
line-encoding ami;
```

To explicitly configure B8ZS line encoding, include the line-encoding statement at the [edit interfaces *interface-name* t1-options] hierarchy level, specifying the b8zs option:

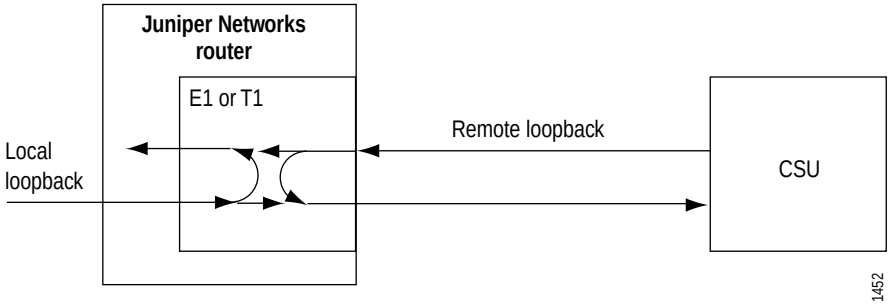
```
[edit interfaces interface-name t1-options]
line-encoding b8zs;
```

When setting the line encoding parameter, you must set the same value for paired ports. Ports 0 and 1 must share the same value, and likewise ports 2 and 3 must share the same value, but ports 0 and 1 can have a different value from that of ports 2 and 3.

Configure E1 and T1 Loopback Capability

You can configure loopback capability between the local E1 or T1 interface and the remote CSU, as shown in Figure 3. You can configure the loopback to be local or remote. With local loopback, the E1 or T1 interface can transmit packets to the CSU, but receives its own transmission back again and ignores data from the CSU. With remote loopback, packets sent from the CSU are received by the E1 or T1 interface but also are immediately retransmitted to the CSU.

Figure 3: Remote and Local E1 or T1 Loopback



To configure loopback capability on an E1 or a T1 interface, include the loopback statement at the [edit interfaces *interface-name* e1-options] or [edit interfaces *interface-name* t1-options] hierarchy level:

```
loopback (local | remote);
```

Packets can be looped on either the local router or the remote CSU. To turn off loopback, remove the loopback statement from the configuration:

```
[edit]
user@host# delete interfaces t1-fpc/pic/port t1-options loopback
```

Configure the E1 and T1 Start and End Flags

By default, an E1 or a T1 interface waits two idle cycles between sending start and end flags. To configure the interface to share the transmission of start and end flags, include the start-end-flag statement at the [edit interfaces *interface-name* e1-options] or [edit interfaces *interface-name* t1-options] hierarchy level, specifying the shared option:

```
start-end-flag shared;
```

To explicitly configure the default of waiting two idle cycles between the start and end flags, include the idle-cycle-flag statement with the filler option:

```
start-end-flag filler;
```

Configure the E1 and T1 Timeslots

To configure the number of timeslots allocated to the interface, include the timeslots statement at the [edit interfaces *interface-name* e1-options] or [edit interfaces *interface-name* t1-options] hierarchy level:

```
timeslots slot-number;
```

The range for the *slot-number* option is 1 through 24 for T1 interfaces and 1 through 32 for E1 interfaces. You can designate any combination of timeslots for usage. The default is to use all timeslots.

To use timeslots 1 through 10, designate *slot-number* as follows:

```
timeslots 1-10;
```

To use timeslots 1 through 5, timeslot 10, and timeslot 24, designate *slot-number* as follows:

```
timeslots 1-5,10,24;
```

To use the first four odd-numbered timeslots, designate *slot-number* as follows:

```
timeslots 1,3,5,7;
```

Spaces are not allowed between timeslot numbers.

For 4-port fractional E1 interfaces only, if you connect to the interface a device that uses timeslot numbering from 0 through 31, you must subtract 1 from the configured number of timeslots. To do this, include the timeslots statement at the [edit interfaces interface-name e1-options] hierarchy level, and offset 1 from the specified slot number.

For example, to use timeslots 3 through 5, timeslot 10, and timeslot 24, designate *slot-number* as follows:

```
[edit interfaces interface-name e1-options]
timeslots 4-6,11,25;
```

The timeslots are offset by 1 to compensate for a device attached to a fractional E1 interface.

Configure E3 and T3 Physical Interface Properties

To configure T3-specific physical interface properties, include the t3-options statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
t3-options {
    bert-algorithm algorithm;
    bert-error-rate rate;
    bert-period seconds;
    (cbit-parity | no-cbit-parity);
    compatibility-mode (digital-link | kentrox | larscom) <subrate value>;
    fcs (32 | 16);
    (feac-loop-respond | no-feac-loop-respond);
    idle-cycle-flag value;
    (long-buildout | no-long-buildout);
    loopback (local | remote);
    (payload-scrambler | no-payload-scrambler);
    start-end-flag value;
}
```

To configure E3-specific physical interface properties, include the e3-options statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
e3-options {
    bert-algorithm algorithm;
    bert-error-rate rate;
    bert-period seconds;
    compatibility-mode (digital-link | kentrox | larscom) <subrate value>;
    fcs (32 | 16);
    idle-cycle-flag value;
    loopback (local | remote);
    (payload-scrambler | no-payload-scrambler);
    start-end-flag value;
}
```

You can configure the following E3-specific and T3-specific properties:

Configure E3 and T3 CSU Compatibility Mode on page 71

Disable T3 C-Bit Parity Mode on page 72

Configure the E3 and T3 Frame Checksum on page 72

Configure E3 and T3 Loopback Capability on page 73

Configure T3 FEAC Response on page 74

Configure the T3 Line Buildout on page 74

Configure the E3 and T3 Idle Cycle Flag on page 75

Configure the E3 and T3 Start and End Flags on page 75

Configure E3 and T3 HDLC Payload Scrambling on page 75

Configure E3 and T3 BERT Properties on page 76

Configure E3 and T3 CSU Compatibility Mode

To configure an E3 or a T3 interface so that it is compatible with the channel service unit (CSU) at the remote end of the line, include the compatibility statement at the [edit interfaces *interface-name* e3-options] or [edit interfaces *interface-name* t3-options] hierarchy level:

```
compatibility-mode (digital-link | kentrox | larscom) <subrate value>;
```

You can configure the interface to be compatible with a Digital Link, Kentrox, or Larscom CSU.

The subrate of an E3 or T3 interface must exactly match that of the remote CSU. To specify the subrate, include the subrate option in the compatibility-mode statement:

For Digital Link CSUs, specify the subrate *value* as the data rate you configured on the CSU in the format *x*kb or *x*.*x*Mb. For a list of specific rate values, use the command completion feature in the CLI. The range is 358 kbps through 33.7 Mbps for E3 interfaces and 301 kbps through 44.2 Mbps for T3 interfaces.

Kentrox CSUs do not support subrate.

For T3 interfaces configured with Larscom CSUs, specify the subrate *value* as a number from 1 through 14 that exactly matches the value configured on the CSU. E3 interfaces do not support the subrate option with Larscom CSUs.

Disable T3 C-Bit Parity Mode

On T3 interfaces only, C-bit parity mode controls the type of framing that is present on the transmitted T3 signal. When C-bit parity mode is enabled, the C-bit positions are used for the FEBE, FEAC, terminal data link, path parity, and mode indicator bits, as defined in ANSI T1.107a-1989. When C-bit parity mode is disabled, the basic T3 framing mode (M13) is used.

By default, C-bit parity mode is enabled. To disable C-bit parity mode and use M13 framing for your T3 link, include the no-cbit-parity statement at the [edit interfaces *interface-name* t3-options] hierarchy level:

```
[edit interfaces interface-name t3-options]
no-cbit-parity;
```

To return to the default, enabling C-bit parity mode, delete the no-cbit-parity statement from the configuration:

```
[edit]
user@host# delete interfaces t3-fpc/pic/port t3-options no-cbit-parity
```

To explicitly enable C-bit parity mode, include the cbit-parity statement at the [edit interfaces *interface-name* t3-options] hierarchy level:

```
[edit interfaces interface-name t3-options]
cbit-parity;
```

Configure the E3 and T3 Frame Checksum

By default, E3 and T3 interfaces use a 16-bit frame checksum. You can configure a 32-bit checksum, which provides more reliable packet verification. However, some older equipment might not support 32-bit checksums.

On a Channelized OC-12 interface, the fcs statement is not supported. To configure FCS on each DS-3 channel, you must include the t3-options fcs statement in the configuration for each channel. For SONET, the Channelized OC-12 interface supports DS-3 to STS-1 to OC-12. For SDH, the Channelized OC-12 interface supports *nx*DS-3 to *nx*VC3 to *nx*AU3 to STM-*n*.

To configure a 32-bit checksum, include the fcs 32 statement at the [edit interfaces *interface-name* e3-options] or [edit interfaces *interface-name* t3-options] hierarchy level:

```
fcs 32;
```

To return to the default 16-bit frame checksum, delete the fcs 32 statement from the configuration:

```
[edit]
user@host# delete interfaces t3-fpc/pic/port t3-options fcs 32
```

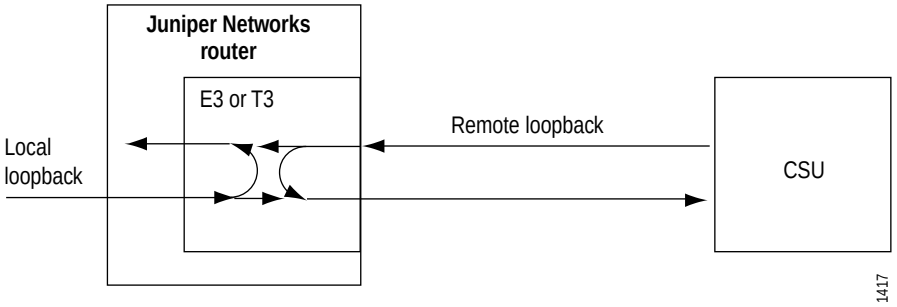
To explicitly configure a 16-bit checksum, include the fcs 16 statement at the [edit interfaces *interface-name* e3-options] or [edit interfaces *interface-name* t3-options] hierarchy level:

```
fcs 16;
```

Configure E3 and T3 Loopback Capability

You can configure loopback capability between the local E3 or T3 interface and the remote CSU, as shown in Figure 4. You can configure the loopback to be local or remote. With local loopback, the E3 or T3 interface can transmit packets to the CSU, but receives its own transmission back again and ignores data from the CSU. With remote loopback, packets sent from the CSU are received by the E3 or T3 interface but also are immediately retransmitted to the CSU.

Figure 4: Remote and Local E3 or T3 Loopback



To configure loopback capability on an E3 or a T3 interface, include the loopback statement at the [edit interfaces *interface-name* e3-options] or [edit interfaces *interface-name* t3-options] hierarchy level:

```
loopback (local | remote);
```

Packets can be looped on either the local router or the remote CSU. To turn off loopback, remove the loopback statement from the configuration:

```
[edit]
user@host# delete interfaces t3-fpc/pic/port t3-options loopback
```

For DS-3 channels on a Channelized OC-12 interface, the SONET loopback statement is supported only for channel 0. It is ignored if included in the configuration for channels 1 through 11. The SONET loopback configured for channel 0 applies to all 12 channels equally. To configure loopbacks on the DS-3 channels, you must include the t3-options loopback statement in the configuration for each channel. Each DS-3 channel can be put in loopback mode independently.

Configure T3 FEAC Response

For T3 interfaces, the T3 far-end alarm and control (FEAC) signal is used to send alarm or status information from the far-end terminal back to the near-end terminal and to initiate T3 loopbacks at the far-end terminal from the near-end terminal. To allow the remote CSU to place the local router into loopback, you must configure the router to respond to the CSU's FEAC request by including the feac-loop-respond statement at the [edit interfaces *interface-name* t3-options] hierarchy level:

```
[edit interfaces interface-name t3-options]
feac-loop-respond;
```

By default, the router does not respond to FEAC requests.

If you have configured remote or local loopback with the T3 loopback statement, the router does not respond to FEAC requests from the CSU even if you have included the feac-loop-respond statement in the configuration. To have the router respond, you must delete the loopback statement from the configuration.

To explicitly configure the router not to respond to FEAC requests, include the no-feac-loop statement at the [edit interfaces *interface-name* t3-options] hierarchy level:

```
[edit interfaces interface-name t3-options]
no-feac-loop-respond;
```

Configure the T3 Line Buildout

A T3 interface has two settings for the T3 line buildout: a short setting, which is less than 225 feet (about 68 meters), and a long setting, which is greater than 225 feet. By default, the interface uses the short setting.

The long-buildout and no-long-buildout statements apply only to copper-cable-based T3 interfaces. You cannot configure a line buildout for a DS-3 channel on a Channelized OC-12 interface, which runs over fiber-optic cable. If you configure this statement on a Channelized OC12 interface, it is ignored.

To have the interface drive a line that is longer than 255 feet, include the long-buildout statement at the [edit interfaces *interface-name* t3-options] hierarchy level:

```
[edit interfaces interface-name t3-options]
long-buildout;
```

To explicitly configure the default short line buildout, include the no-long-buildout statement at the [edit interfaces *interface-name* t3-options] hierarchy level:

```
[edit interfaces interface-name t3-options]
no-long-buildout;
```

Configure the E3 and T3 Idle Cycle Flag

By default, an E3 or a T3 interface transmits the value 0x7E in the idle cycles. To have the interface transmit the value 0xFF (all ones) instead, include the idle-cycle-flag statement at the [edit interfaces *interface-name* e3-options] or [edit interfaces *interface-name* t3-options] hierarchy level, specifying the ones option:

```
idle-cycle-flag ones;
```

To explicitly configure the default value of 0x7E, include the idle-cycle-flag statement with the flags option:

```
idle-cycle-flag flags;
```

Configure the E3 and T3 Start and End Flags

By default, an E3 or a T3 interface waits two idle cycles between sending start and end flags. To configure the interface to share the transmission of start and end flags, include the start-end-flag statement at the [edit interfaces *interface-name* e3-options] or [edit interfaces *interface-name* t3-options] hierarchy level, specifying the shared option:

```
start-end-flag shared;
```

To explicitly configure the default of waiting two idle cycles between the start and end flags, include the start-end-flag statement with the filler option:

```
start-end-flag filler;
```

Configure E3 and T3 HDLC Payload Scrambling

E3 or T3 HDLC payload scrambling, which is disabled by default, provides better link stability. Both sides of a connection must either use or not use scrambling.

On a Channelized OC-12 interface, the SONET payload-scrambler statement is ignored. To configure scrambling on the DS-3 channels on the interface, you can include the payload-scrambler statement at the [edit interfaces *interface-name* e3-options] or [edit interfaces *interface-name* t3-options] hierarchy level for each DS-3 channel:

```
payload-scrambler;
```

To explicitly disable HDLC payload scrambling, include the no-payload-scrambler statement at the [edit interfaces *interface-name* e3-options] or [edit interfaces *interface-name* t3-options] hierarchy level:

```
no-payload-scrambler;
```

To disable payload scrambling again (return to the default), delete the payload-scrambler statement from the configuration:

```
[edit]
user@host# delete interfaces t3-fpc/pic/port t3-options payload-scrambler
```

Configure E3 and T3 BERT Properties

You can configure an E3 or a T3 interface to execute a bit error rate test (BERT) when the interface receives a request to run this test. You specify the duration of the test, the pattern to send in the bit stream, and the error rate to include in the bit stream by including the bert-period, bert-algorithm, and bert-error-rate statements, respectively, at the [edit interfaces *interface-name* e3-options] or [edit interfaces *interface-name* t3-options] hierarchy level:

```
bert-algorithm algorithm;
bert-error-rate rate;
bert-period seconds;
```

seconds is the duration of the BERT procedure, in seconds. The test can last from 1 to 240 seconds; the default is 10 seconds.

rate is the bit error rate. This can be an integer in the range 0 through 7, which corresponds to a bit error rate in the range 10^{-0} (that is, 1 error per bit) to 10^{-7} (that is, 1 error per 10 million bits).

algorithm is the pattern to send in the bit stream. On E3 and T3 interfaces, you can also select the pattern to send in the bit stream by including the bert-algorithm statement at the [edit interfaces *interface-name interface*-options] hierarchy level:

```
[edit interfaces interface-name interface-options]
bert-algorithm algorithm;
```

For a list of supported algorithms, see the CLI possible completions, for example:

```
[edit interfaces t3-0/0/0 t3-options]
user@host# set bert-algorithm ?
Possible completions:
all-ones-repeating   Repeating one bits
all-zeros-repeating  Repeating zero bits
alternating-double-ones-zeros  Alternating pairs of ones and zeros
alternating-ones-zeros  Alternating ones and zeros
pseudo-2e10          Pattern is 2^10 - 1
...
```

See individual interface types for specific hierarchy information. For information about running the BERT procedure, see the *JUNOS Internet Software Operational Mode Command Reference*.

Configure Ethernet Physical Interface Properties

To configure Fast Ethernet-specific physical interface properties, include the fastether-options statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
link-mode (full-duplex | half-duplex);
speed (10m | 100m)
fastether-options {
    802.3ad aex;
    (flow-control | no-flow-control);
    ingress-rate-limit rate;
    (loopback | no-loopback);
    source-address-filter {
        mac-address;
    }
    (source-filtering | no-source-filtering);
}
```

The statement speed (10m | 100m) applies only to the management Ethernet interface (fxp0) and to the Fast Ethernet 12-port and 48-port PICs. The 4-port and 8-port Fast Ethernet PICs support a speed of 100 Mbps only.

To configure Gigabit Ethernet-specific physical interface properties, include the gigether-options statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
gigether-options {
    802.3ad aex;
    (flow-control | no-flow-control);
    (loopback | no-loopback);
    source-address-filter {
        mac-address;
    }
    (source-filtering | no-source-filtering);
}
```

To configure aggregated Ethernet-specific physical interface properties, include the aggregated-ether-options statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
aggregated-ether-options {
    (flow-control | no-flow-control);
    link-speed speed;
    (loopback | no-loopback);
    minimum-links number;
    source-address-filter {
        mac-address;
    }
    (source-filtering | no-source-filtering);
}
```

You can configure the following properties specific to aggregated Ethernet, Fast Ethernet, or Gigabit Ethernet interfaces:

Configure Ethernet Link Aggregation

On Fast Ethernet and Gigabit Ethernet interfaces, you can associate a physical interface with an aggregated Ethernet interface. To enable the aggregated link, include the 802.3ad statement at the [edit interfaces *interface-name* fastether-options] or [edit interfaces *interface-name* gigether-options] hierarchy level:

```
802.3ad aex;
```

You specify the interface instance number *x* to complete the link association; *x* can range from 0 through 15, for a total of 16 aggregated interfaces. You must also include a statement defining ae*x* at the [edit interfaces] hierarchy level. For more information, see “Configure Aggregated Ethernet Interfaces” on page 59. You can optionally specify other physical properties that apply specifically to the aggregated Ethernet interfaces; for details, see “Configure Ethernet Physical Interface Properties” on page 77, and for a sample configuration, see “Example: Configure Aggregated Ethernet Interfaces” on page 250.

The JUNOS software does not support the Link Aggregation Control Protocol (LACP).

Configure Aggregated Ethernet Link Speed

On aggregated Ethernet interfaces, you can set the required link speed for all interfaces included in the bundle. All interfaces that make up a bundle must be the same speed. If you include in the aggregated Ethernet interface an individual link that has a speed different from the speed you specify in the link-speed parameter, an error message will be logged. To set the required link speed, include the link-speed statement at the [edit interfaces *interface-name* aggregated-ether-options] hierarchy level:

```
[edit interfaces interface-name aggregated-ether-options]
link-speed speed;
```

speed can be in bits per second either as a complete decimal number or as a decimal number followed by the abbreviation k (1000), m (1,000,000), or g (1,000,000,000).

Configure Aggregated Ethernet Minimum Links

On aggregated Ethernet interfaces, you can set the minimum number of links that must be up for the bundle as a whole to be labeled up. To set the minimum number, include the minimum-links statement at the [edit interfaces *interface-name* aggregated-ether-options] hierarchy level:

```
minimum-links number;
```

By default, minimum-links has a value of 1. *number* can be a value from 1 through 8.

Configure MAC Address Filtering

On aggregated Ethernet, Fast Ethernet, and Gigabit Ethernet interfaces, you can enable source address filtering, which blocks all incoming packets to that interface. To enable the filtering, include the source-filtering statement at the [edit interfaces *interface-name* aggregated-ether-options], [edit interfaces *interface-name* fastether-options], or [edit interfaces *interface-name* gigether-options] hierarchy level:

```
source-filtering;
```

When source address filtering is enabled, you can configure the interface to receive packets from specific MAC addresses. To do this, specify the MAC addresses in the source-address-filter statement at the [edit interfaces *interface-name* aggregated-ether-options], [edit interfaces *interface-name* fastether-options], or [edit interfaces *interface-name* gigether-options] hierarchy level:

```
source-address-filter {
    mac-address;
    <additional-mac-address;>
}
```

You can specify the MAC address as *nn:nn:nn:nn:nn:nn* or *nnnn.nnnn.nnnn*, where *n* is a hexadecimal number. To specify more than one address, include multiple *mac-address* options in the source-address-filter statement.

If the remote Ethernet card is changed, the interface will not be able to receive packets from the new card because it will have a different MAC address.

Support for source address filters is limited on the Fast Ethernet 12-port and 48-port PIC interfaces.

Configure Loopback Mode

By default, local aggregated Ethernet, Fast Ethernet, or Gigabit Ethernet interfaces connect to a remote system. To place an interface in loopback mode, include the loopback statement at the [edit interfaces *interface-name* aggregated-ether-options], [edit interfaces *interface-name* fastether-options], or [edit interfaces *interface-name* gigether-options] hierarchy level:

```
loopback;
```

To return to the default—that is, to disable loopback mode—delete the loopback statement from the configuration:

```
[edit]
user@host# delete interfaces fe-fpc/pic/port fastether-options loopback
```

To explicitly disable loopback mode, include the no-loopback statement at the [edit interfaces *interface-name* aggregated-ether-options], [edit interfaces *interface-name* fastether-options], or [edit interfaces *interface-name* gigether-options] hierarchy level:

```
[edit interfaces interface-name fastether-options]
no-loopback;
```

Configure Flow Control

By default, the router imposes flow control to regulate the amount of traffic sent out a Fast Ethernet or Gigabit Ethernet interface. This is useful if the remote side of the connection is a Fast Ethernet or Gigabit Ethernet switch.

You can disable flow control if you want the router to permit unrestricted traffic. To disable flow control, include the no-flow-control statement at the [edit interfaces *interface-name* aggregated-ether-options], [edit interfaces *interface-name* fastether-options], or [edit interfaces *interface-name* gigether-options] hierarchy level:

```
[edit interfaces interface-name gigether-options]
no-flow-control;
```

To explicitly reinstate flow control, include the flow-control statement at the [edit interfaces *interface-name* aggregated-ether-options], [edit interfaces *interface-name* fastether-options], or [edit interfaces *interface-name* gigether-options] hierarchy level:

```
[edit interfaces interface-name gigether-options]
flow-control;
```

Configure the Link Characteristics

By default, the router's management Ethernet interface, fxp0, autonegotiates whether to operate in full-duplex or half-duplex mode. Fast Ethernet interfaces can operate in either full-duplex or half-duplex mode, and all other interfaces operate only in full-duplex mode. For Gigabit Ethernet, the link partner must also be set to full duplex.

To explicitly configure an Ethernet interface to operate in either full-duplex or half-duplex mode, include the link-mode statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
link-mode (full-duplex | half-duplex);
```

Configure the Interface Speed

On Fast Ethernet 12-port and 48-port PIC interfaces only, you can explicitly set the interface speed to either 10 Mbps or 100 Mbps.

To explicitly configure the speed on a Fast Ethernet 12-port or 48-port PIC interface, include the speed statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
speed (10m | 100m);
```

Configure the Ingress Rate Limit

On Fast Ethernet 8-port, 12-port, and 48-port PIC interfaces only, you can apply port-based rate limiting to the ingress traffic that arrives at the PIC.

To configure an ingress rate limit on a Fast Ethernet 8-port, 12-port, or 48-port PIC interface, include the ingress-rate-limit statement at the [edit interfaces *interface-name* fastether-options] hierarchy level:

```
[edit interfaces interface-name fastether-options]
ingress-rate-limit rate;
```

rate can range in value from 1 through 100 Mbps.

Configure Multiservice Physical Interface Properties

The passive monitoring PIC is one of a group of multiservice PICs specifically designed to enable IP services. To configure multiservice physical interface properties on the passive monitoring interface, include the multiservice-options statement at the [edit interfaces mo-*fpc*/*pic*/*port*] hierarchy level:

```
[edit interfaces mo-fpc/pic/port]
multiservice-options {
    boot-command filename
    (core-dump | no-core-dump);
    (syslog | no-syslog);
}
```

For more information about the passive monitoring interface, see "Enable Passive Monitoring" on page 115.

Configure SONET/SDH Physical Interface Properties

To configure SONET/SDH physical interface properties, include the sonet-options statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces so-fpc/pic/port]
sonet-options {
    aps {
        advertise-interval milliseconds;
        authentication-key key;
        force;
        hold-time milliseconds;
        lockout;
        neighbor address;
        paired-group group-name;
        protect-circuit group-name;
        request;
        revert-time seconds;
        working-circuit group-name;
    }
    bytes {
        e1-quiet value;
        f1 value;
        f2 value;
        s1 value;
        z3 value;
        z4 value;
    }
    fcs (32 | 16);
    loopback (local | remote);
    path-trace trace-string;
    (payload-scrambler | no-payload-scrambler);
    rfc-2615;
    (z0-increment | no-z0-increment);
}
```

Note that when you configure SONET/SDH OC-48 interfaces for channelized (multiplexed) mode (by including the no-concatenate statement at the [edit chassis fpc *slot-number* pic *pic-number*] hierarchy level), the bytes e1-quiet and bytes f1 options have no effect. The bytes f2, bytes z3, bytes z4, and path-trace options work correctly on channel 0 and work in the transmit direction only on channels 1, 2, and 3.

For DS-3 channels on a Channelized OC-12 interface, the bytes e1-quiet, bytes f1, bytes f2, bytes z3, and bytes z4 options have no effect. The bytes s1 option is supported only for channel 0; it is ignored if configured on channels 1 through 11. The bytes s1 value configured on channel 0 applies to all channels on the interface.

You also can include some of the statements in the sonet-options statement to set SONET/SDH parameters on ATM interfaces.

You can configure the following SONET/SDH physical interface properties:

Configure SONET Header Byte Values

To configure values in SONET header bytes, include the bytes statement at the [edit interfaces *interface-name* sonet-options] hierarchy level:

```
[edit interfaces so-fpc/pic/port sonet-options]
bytes {
    e1-quiet value;
    f1 value;
    f2 value;
    s1 value;
    z3 value;
    z4 value;
}
```

You can configure the following SONET header bytes:

e1-quiet—Default idle byte sent on the orderwire SONET overhead bytes. The router does not support the orderwire channel, and hence sends this byte continuously. For the E1-quiet byte, *value* can be in the range 0 through 255. The default value is 0x7F.

f1, f2, z3, z4—SONET overhead bytes. For these bytes, *value* can be in the range 0 through 255. The default value is 0x00.

s1—Synchronization message SONET overhead byte. This byte is normally controlled as a side effect of the system reference clock configuration and the state of the external clock coming from an interface if the system reference clocks have been configured to use an external reference. For the s1 byte, *value* can be in the range 0 through 255.

On SONET OC-48 interfaces that are configured for channelized (multiplexed) mode (by including the no-concatenate statement at the [edit chassis fpc *slot-number* pic *pic-number*] hierarchy level), the bytes e1-quiet and bytes f1 options have no effect. The bytes f2, bytes z3, bytes z4, and path-trace options work correctly on channel 0 and work in the transmit direction only on channels 1, 2, and 3.

For DS-3 channels on a Channelized OC-12 interface, the bytes e1-quiet, bytes f1, bytes f2, bytes z3, and bytes z4 options have no effect. The bytes s1 option is supported only for channel 0; it is ignored if configured on channels 1 through 11. The bytes s1 value configured on channel 0 applies to all channels on the interface.

Configure SONET z0-increment Option

When configured in SDH framing mode, POS interfaces on a Juniper Networks router might not interoperate with some older versions of ADMs or regenerators that require an incrementing STM ID. To resolve this incompatibility, you can explicitly configure an incrementing STM ID rather than a static one in the SDH overhead by including the z0-increment statement at the [edit interfaces *interface-name* sonet-options] hierarchy level:

```
[edit interfaces so-fpc/pic/port sonet-options]
z0-increment;
```

You should include this statement only for SDH mode; do not use it for SONET mode.

To explicitly disable z0 incrementing, include the no-z0-increment statement at the [edit interfaces so-fpc/pic/port sonet-options] hierarchy level:

```
[edit interfaces so-fpc/pic/port sonet-options]
no-z0-increment;
```

Current SDH standards specify a set of 3*n* overhead bytes in an STM-*n* that includes the J0 section trace byte. The rest are essentially unused (spare Z0) and contain hex values (0x01, 0xCC, 0xCC ... 0xCC).

The older version of the standard specified that the same set of bytes should contain an incrementing sequence: 1, 2, 3, ..., 3*n*. Their use was still unspecified, although they might have been used to assist in frame alignment. The z0-increment option enables Juniper Networks routers to interoperate with older equipment that relies on those bytes for frame alignment.

The STM identifier has a precise definition from the SDH specs. In ITU-T Recommendation G.707, *Network node interface for the synchronous digital hierarchy (SDH)* (03/96), section 9.2.2.2:

> NOTE: STM identifier: C1
>
> In earlier versions of the Recommendation, the content of bytes located at S (1, 7, 1) or [1, 6N + 1] to S (1,7, N) or [1, 7N] was defined as a unique identifier indicating the binary value of the multi-column, interleave depth coordinate, c. It may have been used to assist in frame alignment.

Configure the SONET Frame Checksum

By default, SONET interfaces use a 16-bit frame checksum. You can configure a 32-bit checksum, which provides more reliable packet verification. However, some older equipment may not support 32-bit checksums.

To configure a 32-bit checksum, include the fcs statement at the [edit interfaces *interface-name* sonet-options] hierarchy level:

```
[edit interfaces so-fpc/pic/port sonet-options]
fcs 32;
```

To return to the default 16-bit frame checksum, delete the fcs 32 statement from the configuration:

```
[edit]
user@host# delete interfaces so-fpc/pic/port sonet-options fcs 32
```

To explicitly configure a 16-bit checksum, include the fcs statement at the [edit interfaces *interface-name* sonet-options] hierarchy level:

```
[edit interfaces so-fpc/pic/port sonet-options]
fcs 16;
```

On a Channelized OC-12 interface, the sonet-options fcs statement is not supported. To configure FCS on each DS-3 channel, you must include the t3-options fcs statement in the configuration for each channel.

Configure SONET Loopback Capability

To configure loopback capability on a SONET interface, include the loopback statement at the [edit interfaces *interface-name* sonet-options] hierarchy level:

```
[edit interfaces so-fpc/pic/port sonet-options]
loopback (local | remote);
```

Packets can be looped on either the local or the remote router. To turn off loopback, remove the loopback statement from the configuration:

```
[edit]
user@host# delete interfaces so-fpc/pic/port sonet-options loopback
```

For DS-3 channels on a Channelized OC-12 interface, the SONET loopback statement is supported only for channel 0; it is ignored if included in the configuration for channels 1 through 11. The SONET loopback configured for channel 0 applies equally to all 12 channels.

You can configure 12 channels per interface, and each interface can have logical interfaces. To configure loopbacks on the DS-3 channels, you must include the t3-options loopback statement in the configuration for each channel. Each DS-3 channel can be put in loopback mode independently.

Configure the SONET Path Trace Identifier

The SONET path trace identifier is a text string that identifies the circuit. If the string contains spaces, enclose it in quotation marks. The common convention is to use the circuit identifier as the path trace identifier. If you do not configure an identifier, the JUNOS software uses the system and interface names. The local system's path trace identifier is displayed when a show interfaces command is issued on the remote system.

For DS-3 channels on a Channelized OC-12 interface, you can configure a unique path trace for each of the 12 channels. Each path trace can be up to 16 bytes.

To configure a path trace identifier, include the path-trace statement at the [edit interfaces *interface-name* sonet-options] hierarchy level:

```
[edit interfaces so-fpc/pic/port sonet-options]
path-trace trace-string;
```

Configure SONET HDLC Payload Scrambling

SONET HDLC payload scrambling, which is enabled by default, improves link stability. Both sides of a connection must either use or not use scrambling.

On a Channelized OC-12 interface, the SONET payload-scrambler statement is ignored. To configure scrambling on the DS-3 channels on the interface, you can include the t3-options payload-scrambler statement in the configuration for each DS-3 channel.

To disable HDLC payload scrambling, include the no-payload-scrambler statement at the [edit interfaces *interface-name* sonet-options] hierarchy level:

```
[edit interfaces so-fpc/pic/port sonet-options]
no-payload-scrambler;
```

To return to the default, that is, to re-enable payload scrambling, delete the no-payload-scrambler statement from the configuration:

```
[edit]
user@host# delete interfaces so-fpc/pic/port sonet-options no-payload-scrambler
```

To explicitly enable payload scrambling, include the payload-scrambler statement at the [edit interfaces *interface-name* sonet-options] hierarchy level:

```
[edit interfaces so-fpc/pic/port sonet-options]
payload-scrambler;
```

Configure SONET RFC 2615 Support

RFC 2615 requires certain C2 byte and FCS settings in addition to the default values configured in accordance with RFC 1619.

To enable support for the RFC 2615 features, include the rfc-2615 statement at the [edit interfaces *interface-name* sonet-options] hierarchy level:

```
[edit interfaces so-fpc/pic/port sonet-options]
rfc-2615;
```

Configure APS

Automatic Protection Switching (APS) is used by SONET add/drop multiplexers (ADMs) to protect against circuit failures. The JUNOS implementation of APS allows you to protect against circuit failures between an ADM and one or more routers, and between multiple interfaces in the same router. When a circuit or router fails, a backup immediately takes over.

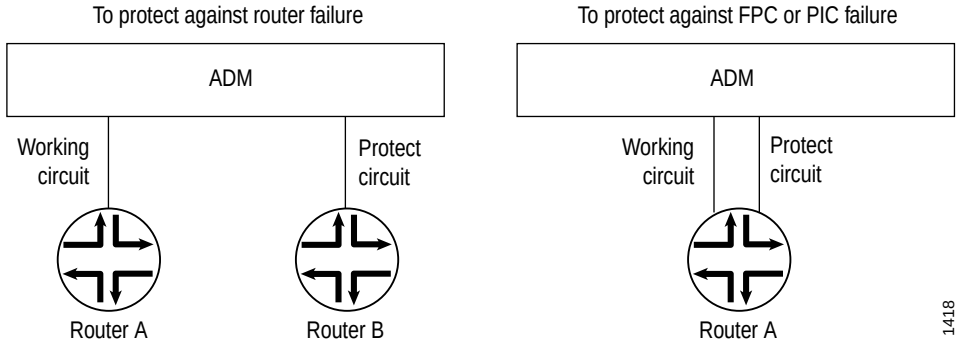
The JUNOS software supports APS 1 + 1 switching, bidirectional only, and either revertive or nonrevertive mode. The JUNOS software does not transmit identical data on the working and protect circuits, as the APS specification requires for 1 + 1 switching, but this causes no operational impact.

For DS-3 channels on a Channelized OC-12 interface, you can configure APS on channel 0 only. If you configure APS on channels 1 through 11, it is ignored.

With APS, you configure two circuits, a *working circuit* and a *protect circuit*, as shown in Figure 5. Normally, traffic is carried on the working circuit (that is, the working circuit is the active circuit), and the protect circuit is disabled. If the working circuit fails or degrades, or if the working router fails, the ADM and the protect router switch the traffic to the protect circuit, and the protect circuit becomes the active circuit.

To configure APS, you configure a working and a protect circuit. To protect against a router failure, you connect two routers to the ADM, configuring one of them as the working router and the second as the protect router. To protect against a PIC or an FPC failure, you connect one router to the ADM through both the working and protect circuits, configuring one of the PICs or FPCs as the working circuit and the second as the protect circuit.

Figure 5: APS Configuration Topologies



To configure APS, include the aps statement at the [edit interfaces *interface-name* sonet-options] hierarchy level:

```
[edit interfaces interface-name sonet-options]
aps {
    advertise-interval milliseconds;
    authentication-key key;
    force;
    hold-time milliseconds;
    lockout;
    neighbor address;
    paired-group group-name;
    protect-circuit group-name;
    request;
    revert-time seconds;
    working-circuit group-name;
}
```

You can configure the following APS properties:

Configure Basic APS Support on page 88

Configure Switching between the Working and Protect Circuits on page 90

Configure Revertive Mode on page 91

Configure APS Timers on page 91

Configure APS Load Sharing between Circuit Pairs on page 92

Configure Basic APS Support

To set up a basic APS configuration, configure one interface to be the working circuit and a second to be the protect circuit. If you are using APS to protect against router failure, configure one interface on each router. If you are using APS to protect against FPC failure, configure two interfaces on the router, one on each FPC.

For each working-protect circuit pair, configure the following:

Group name—Creates the association between the two circuits. Configure the same group name for both the working and protect routers.

Authentication key—If you are configuring one router to be the working router and a second to be the protect router, you configure this on both interfaces. Configure the same key for both the working and protect routers.

If you configure APS on the same router for FPC and PIC failure protection, the authentication-key statement is not needed.

Address of the other interface on the other router—If you are configuring one router to be the working router and a second to be the protect router, you must configure the address of the remote interface.

The address you specify for the neighbor must never be routed through the interface on which APS is configured, because it might cause network instability. We strongly recommend that you directly connect the working and protect routers and that you configure the interface address of each end's directly connected interface IP as the neighbor address.

If you configure APS on the same router for FPC and PIC failure protection, the address statement is not needed.

The working and protect configurations on the routers must match the circuit configurations on the ADM; that is, the working router must be connected to the ADM's working circuit, and the protect router must be connected to the protect circuit.

To set up a basic APS configuration, include the following statements at the [edit interfaces *interface-name* sonet-options] hierarchy level:

On the working router/circuit:

```
[edit interfaces so-fpc/pic/port sonet-options]
aps {
    working-circuit group-name;
    authentication-key key; # Include only if working circuit is on a different router
    neighbor address; # Include only if protect circuit is on a different router
}
```

On the protect router/circuit:

```
aps {
    protect-circuit group-name;
    authentication-key key; # Include only if working circuit is on a different router
    neighbor address;# Include only if working circuit is on a different router
}
```

For example, configure Router A to be the working router and Router B to be the protect router.

On Router A (the working router):

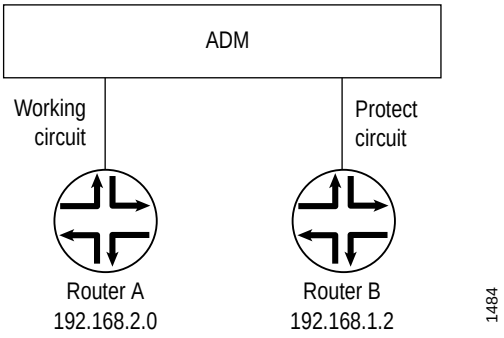
```
[edit interfaces so-6/1/1 sonet-options]
aps {
    working-circuit San-Jose;
    authentication-key "$9$B2612345";
    neighbor 192.168.1.2;# address of Router B's interface on the link between A and B
}
```

On Router B (the protect router):

```
[edit interfaces so-0/0/0 sonet-options]
aps {
    protect-circuit San-Jose;
    authentication-key "$9$B2612345";
    neighbor 192.168.2.0;# address of Router A's interface on the link between A and B
}
```

In this example, Router A's interface IP address is the neighbor address configured on Router B, and Router B's interface IP address is the neighbor address configured on Router A, as shown in Figure 6.

Figure 6: APS Example



As a second example, configure one interface on a router to be the working circuit and another interface to be the protect circuit.

On Router A:

```
[edit interfaces so-2/1/1 sonet-options]
aps {
    working-circuit Hayward;
}
[edit interfaces so-3/0/2 sonet-options]
aps {
    protect-circuit Hayward;
}
```

Configure Switching between the Working and Protect Circuits

When there are multiple reasons to switch between the working and protect circuits, a priority scheme is used to decide which circuit to use. The routers and the ADM might automatically switch traffic between the working and protect circuits because of circuit and router failures. You can also choose to switch traffic manually between the working and protect circuits. There are three priority levels of manual configuration, listed here in order from lowest to highest priority:

Request (also known as manual switch)—Overridden by signal failures, signal degradations, or any higher-priority reasons.

Force (also known as forced switch)—Overrides manual switches, signal failures, and signal degradation.

Lockout (also known as lockout of protection)—Do not switch between the working and protect circuits.

A router failure is considered to be equivalent to a signal failure on a circuit.

To perform a manual switch, include the request statement at the [edit interfaces *interface-name* sonet-options aps] hierarchy level:

```
[edit interfaces so-fpc/pic/port sonet-options aps]
request (protect | working);
```

This statement takes effect only if there are no higher-priority reasons to switch.

When the working circuit is operating in nonrevertive mode, use the request working statement to switch the circuit manually to being the working circuit or to override the revert timer.

To perform a forced switch, include the force statement at the [edit interfaces *interface-name* sonet-options aps] hierarchy level:

```
[edit interfaces so-fpc/pic/port sonet-options aps]
force (protect | working);
```

This statement is honored only if there are no higher-priority reasons to switch. This configuration can be overridden by a signal failure on the protect circuit, causing a switch to the working circuit.

To configure a lockout of protection, forcing the use of the working circuit and locking out the protect circuit regardless of anything else, include the lockout statement at the [edit interfaces *interface-name* sonet-options aps] hierarchy level:

```
[edit interfaces so-fpc/pic/port sonet-options aps]
lockout;
```

Configure Revertive Mode

By default, APS is nonrevertive, which means that if the protect circuit becomes active, traffic is not switched back to the working circuit unless the protect circuit fails or you manually configure a switch to the working circuit. In revertive mode, traffic is automatically switched back to the working circuit.

You should configure the ADM and routers consistently with regard to revertive or nonrevertive mode.

To configure revertive mode, include the revert-time statement, specifying the amount of time to wait after the working circuit has again become functional before making the working circuit active again:

```
[edit interfaces so-fpc/pic/port sonet-options aps]
revert-time seconds;
```

If you are using nonrevertive APS, you can use the request working statement to switch the circuit manually to being the working circuit or to override the revert timer (configured with the revert-time statement).

Configure APS Timers

The protect and working routers periodically send packets to their neighbors to advertise that they are operational. By default, these advertisement packets are sent every 1000 milliseconds. A router considers its neighbor to be operational for a period called the *hold time* ; by default, this period is equal to three times the advertisement interval. If the protect router does not receive an advertisement packet from the working router within the hold time configured on the protect router, the protect router assumes that the working router has failed and becomes active.

APS is symmetric; either side of a circuit can time out the other side (for example, when detecting a crash of the other). Under normal circumstances, the failure of the protect router does not cause any changes because the traffic is already moving on the working router. However, if you had configured request protect and the protect router failed, the working router would enable its interface.

To modify the advertisement interval, include the advertise-interval at the [edit interfaces *interface-name* sonet-options aps] hierarchy level:

```
[edit interfaces so-fpc/pic/port sonet-options aps]
advertise-interval milliseconds;
```

To modify the hold time, include the hold-time at the [edit interfaces *interface-name* sonet-options aps] hierarchy level:

```
[edit interfaces so-fpc/pic/port sonet-options aps]
hold-time milliseconds;
```

The advertisement intervals and hold times on the protect and working routers can be different.

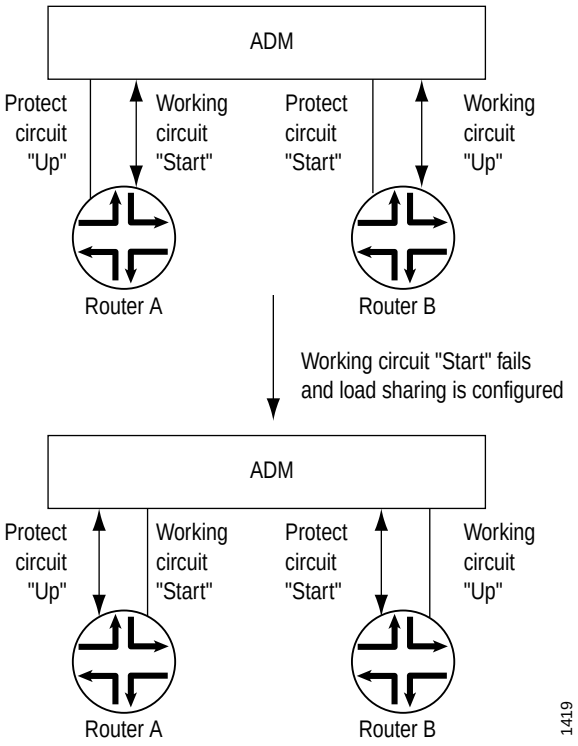
Configure APS Load Sharing between Circuit Pairs

When two routers are connected to a single ADM, you can have them back up each other on two different pairs of circuits. This arrangement provides load balancing between the routers if one of the working circuits fails.

Figure 7 illustrates load sharing between circuits on two routers. Router A has a working circuit “Start” and a protect circuit “Up,” and Router B has a working circuit “Up” and a protect circuit “Start.” Under normal circumstances, Router A carries the “Start” circuit traffic and Router B carries the “Up” circuit traffic. If the working circuit “Start” were to fail, Router B would end up carrying all the traffic for both the “Start” and “Up” circuits.

To balance the load between the circuits, you pair the two circuits. In this case, you pair the “Start” and “Up” circuits. Then, if the working circuit “Start” fails, the two routers automatically switch the “Up” traffic from the working to the protect circuit so that each router is still carrying only one circuit’s worth of traffic. That is, the working circuit on Router A would be “Up” and the working circuit on Router B would be “Start.”

Figure 7: APS Load Sharing between Circuit Pairs



To configure load sharing between two working-protect circuit pairs, include the paired-group statement at the [edit interfaces so-*fpc*/*pic*/*port* sonet-options aps] hierarchy level:

```
[edit interfaces so-fpc/pic/port sonet-options aps]
paired-group group-name;
```

In this statement, *group-name* is the name of the group you assigned to one of the circuits with the working-circuit and protect-circuit statements. The software automatically configures the remainder of the load-sharing setup based on the group name.

Example: Configure APS Load Sharing between Circuit Pairs

Configure APS load sharing to match the configuration shown in Figure 7.

On Router A:

```
[edit interfaces so-7/0/0 sonet-options aps]
user@host# set working-circuit start
[edit interfaces so-7/0/0 sonet-options aps]
user@host# set authentication-key linsey
[edit interfaces so-7/0/0 sonet-options aps]
user@host# set paired-group "Router A-Router B"
...
[edit interfaces so-0/0/0 sonet-options aps]
user@host# set protect-circuit up
[edit interfaces so-0/0/0 sonet-options aps]
user@host# set authentication-key woolsey
[edit interfaces so-0/0/0 sonet-options aps]
user@host# set paired-group "Router A-Router B"
```

On Router B:

```
[edit interfaces so-1/0/0 sonet-options aps]
user@host# set working-circuit up
[edit interfaces so-1/0/0 sonet-options aps]
user@host# set authentication-key woolsey
[edit interfaces so-1/0/0 sonet-options aps]
user@host# set paired-group "Router A-Router B"
...
[edit interfaces so-6/0/0 sonet-options aps]
user@host# set protect-circuit start
[edit interfaces so-6/0/0 sonet-options aps]
user@host# set authentication-key linsey
[edit interfaces so-6/0/0 sonet-options aps]
user@host# set paired-group "Router A-Router B"
```

Configure Aggregated SONET/SDH Interfaces

The JUNOS software enables link aggregation of SONET/SDH interfaces; this is similar to Ethernet link aggregation, but is not defined in a public standard. You configure an aggregated SONET/SDH virtual link by specifying the link number as a physical device and then associating a set of physical interfaces that have the same speed.

The JUNOS software does not provide load balancing for multicast traffic on aggregated interfaces. If a link carrying multicast data goes down, another link carries the traffic. This provides redundancy, not more bandwidth.

By default, no aggregated SONET/SDH interfaces are created. You must define the number of aggregated SONET/SDH interfaces by including the device-count statement at the [edit chassis aggregated-devices sonet] hierarchy level:

```
[edit chassis aggregated-devices sonet]
device-count number;
```

The maximum number of aggregated interfaces is 16, and the assigned number can range from 0 through 15. You should not mix SONET and SDH mode on the same aggregated interface. For more information, see the *JUNOS Internet Software Guide: Getting Started*.

SONET aggregation is proprietary to the JUNOS software and might not work with other software.

To configure aggregated SONET/SDH interfaces, assign a number for the aggregated SONET/SDH interface as*x* at the [edit interfaces] hierarchy level:

```
[edit interfaces]
asx {
...
}
```

The following example shows an aggregated SONET/SDH configuration:

```
[edit interfaces]
as0 {
    aggregated-sonet-options {
        minimum-links 1;
        link-speed oc3;
    }
    unit 0 {
        family inet {
            address 10.2.11.1/32 {
                destination 10.2.11.3;
            }
        }
    }
}
```

You also need to specify the constituent physical interfaces by including the aggregate statement at the [edit interfaces *interface-name* sonet-options] hierarchy level; for more information, see "Configure SONET Link Aggregation" on page 96. You can optionally specify other physical properties that apply specifically to the aggregated SONET/SDH interfaces; for details, see "Configure SONET/SDH Physical Interface Properties" on page 82. For a sample configuration, see "Example: Configure Aggregated SONET/SDH Interfaces" on page 294.

To remove the configuration statements related to as*x* and set the aggregated SONET/SDH interface to down state, delete the interface from the configuration:

```
[edit]
user@host# delete interfaces asx
```

However, the aggregated SONET/SDH interface is not deleted until you delete the chassis aggregated-devices sonet device-count configuration statement.

You can configure the following aggregated SONET/SDH properties:

Configure SONET Link Aggregation on page 96

Configure Aggregated SONET Link Speed on page 96

Configure Aggregated SONET Minimum Links on page 97

Configure Filters or Sampling on Aggregated SONET Links on page 97

Configure SONET Link Aggregation

On SONET/SDH interfaces, you can associate a physical interface with an aggregated SONET interface. To associate the interface with an aggregated SONET link, include the aggregate statement at the [edit interfaces *interface-name* sonet-options] hierarchy level:

```
[edit interfaces interface-name sonet-options]
aggregate asx;
```

x is the interface instance number and can range from 0 through 15, for a total of 16 aggregated interfaces. You should not mix SONET and SDH mode on the same aggregated interface. You must also include a statement defining as*x* at the [edit interfaces] hierarchy level. For a sample configuration, see “Example: Configure Aggregated SONET/SDH Interfaces” on page 294.

SONET aggregation is proprietary to the JUNOS software and might not work with other software.

You can combine like interfaces only, so each physical interface in the aggregate must be the same speed.

Configure Aggregated SONET Link Speed

On aggregated SONET interfaces, you can set the required link speed for all interfaces included in the bundle. All interfaces that make up a bundle must be the same speed. If you include in the aggregated SONET interface an individual link that has a speed different from the speed you specify in the link-speed parameter, an error message will be logged. To set the required link speed, include the link-speed statement at the [edit interfaces *interface-name* aggregated-sonet-options] hierarchy level:

```
[edit interfaces interface-name aggregated-sonet-options]
link-speed speed;
```

speed can be one of the following values:

oc3—Links are OC-3c or STM-1c.

oc12—Links are OC-12c or STM-4c.

oc48—Links are OC-48c or STM-16c.

oc192—Links are OC-192c or STM-64c.

Configure Aggregated SONET Minimum Links

On aggregated SONET interfaces, you can set the minimum number of links that must be up for the bundle as a whole to be labeled up. To set the minimum number, include the minimum-links statement at the [edit interfaces *interface-name* aggregated-sonet-options] hierarchy level:

```
[edit interfaces interface-name aggregated-sonet-options]
minimum-links number;
```

By default, minimum-links has a value of 1. *number* can be a value from 1 through 8.

Configure Filters or Sampling on Aggregated SONET Links

To set up firewall filters or sampling on aggregated SONET interfaces, you must configure the as*x* interface with these properties. The filters function in the same manner as on other interfaces.

To configure a filter on the as*x* interface, include the filter statement at the [edit interfaces as*x*] hierarchy level:

```
[edit interfaces asx]
filter {
    input input-filter-name;
    output output-filter-name;
}
```

To define properties of firewall filters, include one or more filter statements at the [edit firewall] hierarchy level:

```
[edit firewall]
filter filter-name {
    term term-name {
        from {
            match-conditions;
        }
        then {
            action;
            action-modifiers;
        }
    }
}
```

For more information, see the *JUNOS Internet Software Configuration Guide: Policy Framework*.

You configure sampling on aggregated SONET/SDH interfaces in a similar way, as shown in this example:

```
[edit interfaces]
asx {
    unit 0 {
        family inet {
            address 10.2.11.1/32 {
                destination 10.2.11.3;
            }
            filter {
                input input-sampler-name;
            }
        }
    }
}
```

To define the sampling filter and the forwarding action, include the filter statement at the [edit firewall] hierarchy level and the sampling statement at the [edit forwarding-options] hierarchy level:

```
[edit firewall]
filter input-sampler-name {
    term match-any-input {
        then {
            sample;
            accept;
        }
    }
}

[edit forwarding-options]
sampling {
    input {
        family inet {
            rate number;
            run-length number;
        }
    }
}
```

For more information, see the *JUNOS Internet Software Configuration Guide: Policy Framework*.

Configure 802.1Q VLAN Tagging

For Ethernet, Fast Ethernet, and Gigabit Ethernet interfaces, the software supports a subset of the IEEE 802.1Q standard for channelizing an Ethernet interface into multiple logical interfaces, allowing many hosts to be connected to the same Gigabit Ethernet switch, but preventing them from being in the same routing or broadcast domain.

The software supports receiving and forwarding routed Ethernet frames with 802.1Q VLAN tags and running VRRP over 802.1Q-tagged interfaces. To configure the router to receive and forward frames with 802.1Q VLAN tags, include the vlan-tagging statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
vlan-tagging;
```